Image Space Tensor Field Visualization using a LIC-like Method

Sebastian Eichelbaum[1] Mario Hlawitschka[2] Gerik Scheuermann[1]

[1] Abteilung für Bild- und Signalverarbeitung, Institut für Informatik, Universität Leipzig
[2] Institute for Data Analysis and Visualization (IDAV), and Department of Computer Science, University of California, Davis

Outline

1. Introduction
   - Focus
   - Tensor field visualization
   - Motivation and Goals
2. The method
   - Step 0: Input
   - Step 1: Projection to Image Space
   - Step 2: Silhouette detection
   - Step 3: Advection
   - Step 4: Compositing
3. Results
4. Problems and Further Work

Outline

1. Introduction
   - Focus
   - Tensor field visualization
   - Motivation and Goals
2. The method
   - Step 0: Input
   - Step 1: Projection to Image Space
   - Step 2: Silhouette detection
   - Step 3: Advection
   - Step 4: Compositing
3. Results
4. Problems and Further Work

Focusing on ...

- Second-order tensor fields
- Diffusion tensors
  - positive definite
  - symmetric
  - three orthogonal eigenvectors without orientation
- Medical DTI visualization, but not limited to

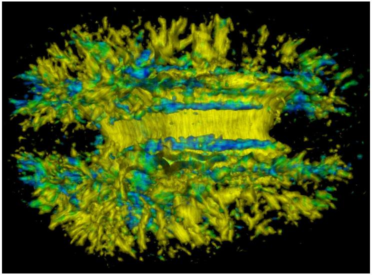
Tensor field visualization

- Inspired by methods used in Scalar- and Vector field visualization
- Often using derived metrics
- Common methods:
  - Colormaps
  - HyperLIC (Zheng et al. [ZP03])
  - Tensor Glyphs (Kindlmann [Kin04])
  - Direct Volume Rendering (foundations in [Bli82, KVH84])
  - Advection Diffusion Tensorlines (Kindlmann et al. [WKL99])
  - Hyperstreamlines (Delmarcelle et al. [DH92])

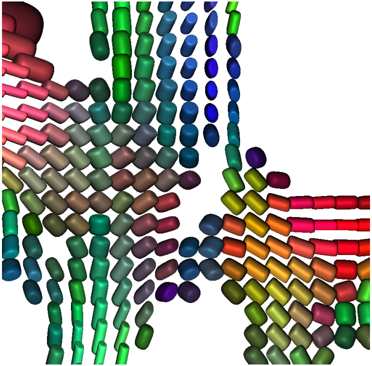
Weaknesses and Problems

- Limitation to local or global data representation
  - no smooth and interactive transition between levels of detail
- Severe limitations in data size
- Interactive performance
- Limitations in number of represented tensor attributes

Examples I

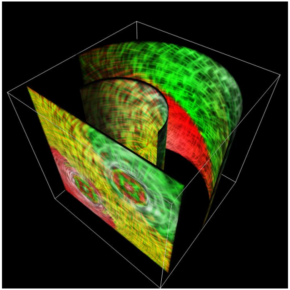


(a) HyperLIC

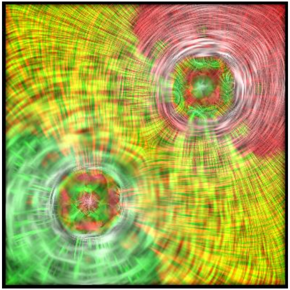


(b) Superquadrics

Figure: HyperLIC and Superquadric Tensor Glyphs



(a) Method applied to several surfaces



(b) XZ-Slice

Figure: Hotz et al. [HFHJ09]. Limitation to type of surface.

Motivation and Goals

- Easy perceptibility of structures
  - More bold representation of diffusion structures
- Continuous perception of structures during transformation
  - Often problematic with image space based methods
- Allow smooth transition between local and global structures
- Realtime ability
- Applicability on arbitrary geometry

Outline

1 Introduction
- Focus
- Tensor field visualization
- Motivation and Goals

2 The method
- Step 0: Input
- Step 1: Projection to Image Space
- Step 2: Silhouette detection
- Step 3: Advection
- Step 4: Compositing

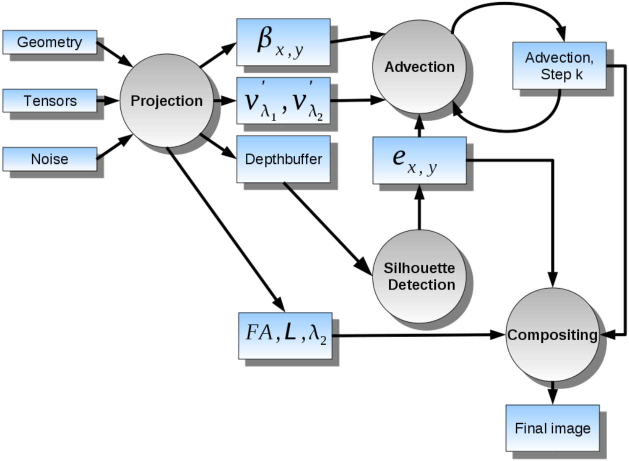
3 Results

4 Problems and Further Work

Overview I

- Move problem to image space
- Divide into small parallelizable parts
  - Utilize GPU parallelism
- Implementation using OpenGL, GLSL and Framebuffer Objects
  - But smaller float precision
  - Many limitations
- Textures as transport media

Overview II



Geometry
Tensors
Noise
Projection
$\beta_{x,y}$
$v'_{\lambda_1}, v'_{\lambda_2}$
Depthbuffer
Advection
Advection, Step k
$e_{x,y}$
Silhouette Detection
FA, L, λ_2
Compositing
Final image

Noise

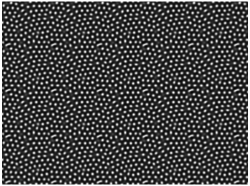


Figure: Tiled 100x100 pixel reaction diffusion texture with $D_a = 0.125$ and $D_b = 0.031$.

- Initial calculation of input noise
  - Reaction Diffusion ([Tur52])
  - Create once, reuse every pass
- Since computational expensive: tiling

Geometry and Tensors

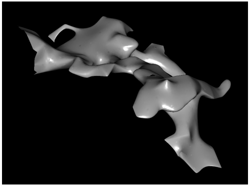
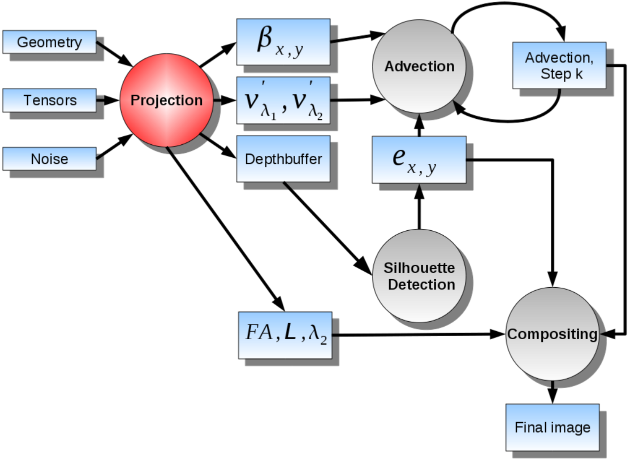


Figure: Input geometry with Phong lighting.

- Geometry calculated using arbitrary metric and algorithm
- Tensors uploaded as two 3D texture coordinates
- Requirements to geometry:
  - Smooth normals
  - Not self-intersecting

Step 1: Projection to Image Space



Tensor projection I

- Tensor interpolated using GPU
- Projection to geometry surface ($n = s \cdot v_{\lambda_3}$):

$$T' = P \cdot T \cdot P^T \text{ mit } P = \begin{pmatrix} 1 - n_x^2 & -n_y n_x & -n_z n_x \\ -n_x n_y & 1 - n_y^2 & -n_z n_y \\ -n_x n_z & -n_y n_z & 1 - n_z^2 \end{pmatrix}.$$

- Eigenvalue decomposition using Hasan et al. [HBPA01]
  - Eigenvalues: λ_i with $i \in \{1, 2\}$
  - Eigenvectors: v_{λ_i} with $i \in \{1, 2\}$
- Eigenvectors still in geometries object coordinate system

Tensor projection II

- Projection to image space using OpenGL's Modelviewmatrix M_M and Projectionmatrix M_P: $v'_{\lambda_i} = M_P \times M_M \times v_{\lambda_i}$, with $(i \in 1, 2)$ and $v'_{\lambda_i} \in \mathbb{R}^2$
- May not need to be orthogonal anymore $(< v'_{\lambda_1}, v'_{\lambda_2} > \neq 0)$
- Scale to $[0, 1]$: $v''_{\lambda_i} = \frac{1}{2} + \frac{1}{2} * \frac{v'_{\lambda_i}}{\|v'_{\lambda_i}\|_\infty}$ with $i \in \{1, 2\}$ and $\|v'_{\lambda_i}\|_\infty \neq 0$

Noise Texture Mapping I

- To ensure consistency during transformation
- Many methods available
  - Texture Atlases ([PCK04, IOK00])
  - Reaction Diffusion directly on the geometry ([Tur91])
  - 3D textures ([WE04])
  - Mostly computational expensive or geometry dependent results
- Own heuristics developed
  - Not C^1 constant
  - May introduce minor distortions
  - Allows seamless scaling
  - Good trade-off between computation time and visual quality

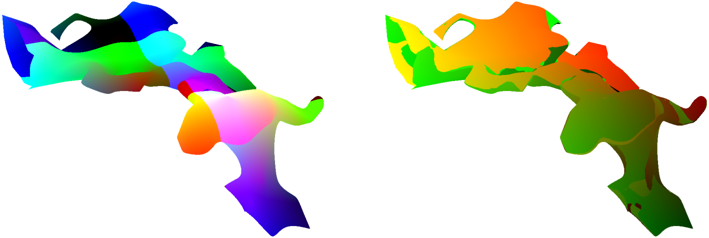
Noise Texture Mapping II

- Transformation of vertex v_g to *voxelized* space:

$$v_{voxel} = v_g \cdot \begin{pmatrix} l & 0 & 0 & -b_{min_x} \\ 0 & l & 0 & -b_{min_y} \\ 0 & 0 & l & -b_{min_z} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- l is its size, b_{min} origin of voxel space in world coordinates
- Discretize to voxels borders $v_{hit} = v_{voxel} - \lfloor v_{voxel} \rfloor$
- Texture coordinate t is then defined as:
  $t = (v_{hit_i}, v_{hit_j})$, with $i \neq j \neq k \wedge (n_k = max\{n_i, n_j, n_k\})$

Noise Texture Mapping III)



(a) v_{hit} (b) t

Figure: Illustration of v_{hit} and t for illustration.

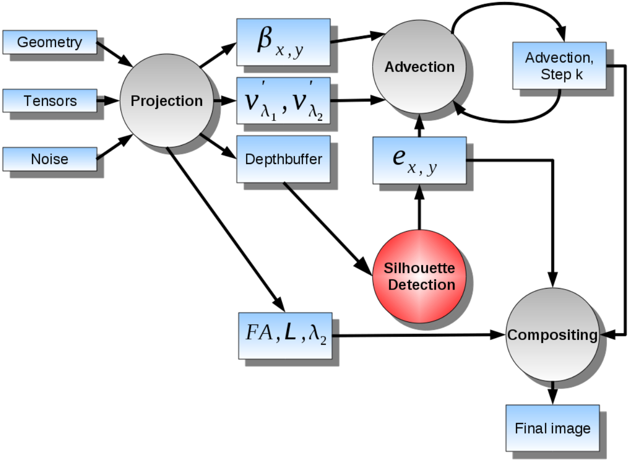
Noise Texture Mapping IV

Figure: Noise mapped to surface ($\beta_{i,j}$).

Further calculations

- Phong intensity $\mathcal{L}$
- Mean Diffusivity
- Fractional Anisotropy
- Colormapping: $c^{FA \cdot v_{\lambda_i}}(T) = \frac{|v_{\lambda_i}|}{\|v_{\lambda_i}\|} * FA(T)$

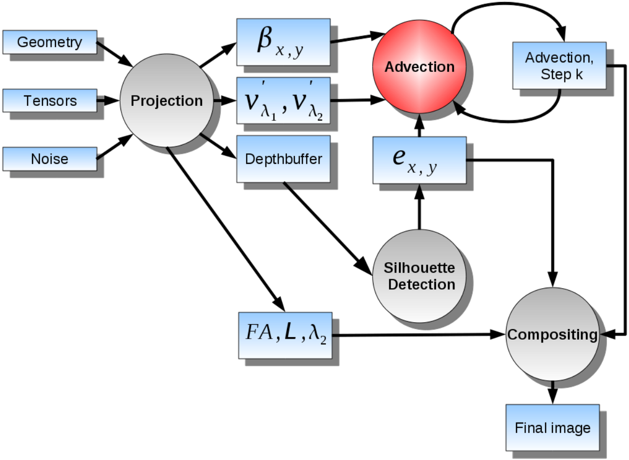
Step 2: Silhouette detection I



Step 2: Silhouette detection II

- Creation of silhouette texture
  $e : (x, y) \rightarrow s$, with $x, y, s \in [0, 1]$ using depthbuffer
- Fold using Laplace filter kernel: $D^2_{xy} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

Step 3: Advection I



Step 3: Advection II

- Access to discrete, intermediate LIC-textures P^{λ_1} and P^{λ_2} using: $f_P : (x, y) \to p$, with $x, y, p \in [0, 1]$
  - Interpolation
- Iteration on both textures for each pixel:

$$\forall x, y \in [0, 1] : \forall \lambda \in \{\lambda_1, \lambda_2\} :$$

$$p_0^\lambda = \beta_{x,y},$$

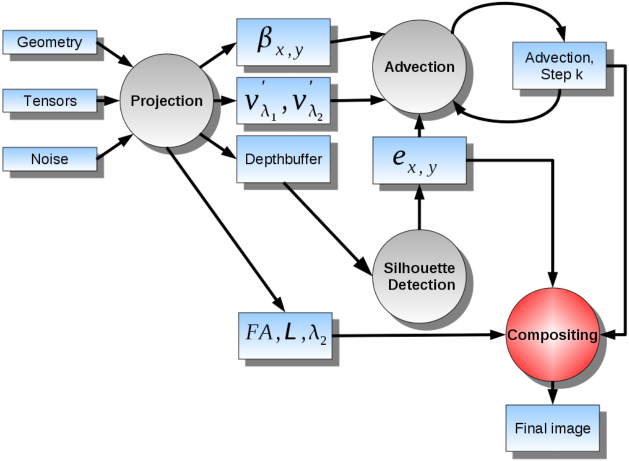
$$p_{i+1}^\lambda = k \cdot \beta_{x,y} + (1 - k) \cdot \frac{f_{p_i^\lambda}(x + v'_{\lambda_x}, y + v'_{\lambda_y}) + f_{p_i^\lambda}(x - v'_{\lambda_x}, y - v'_{\lambda_y})}{2}.$$

  - k describes "roughness" (the smaller k is, the more smooth the final image looks)
  - Advection needs to be done in both directions, since eigenvectors do not have an orientation
  - Stop iteration if $|p_i^\lambda - p_{i+1}^\lambda| < \epsilon$

Step 3: Advection III

Figure: Advection of Eigenvector field v'_{λ_1} after 10 iterations.

Step 4: Compositing I



Step 4: Compositing II

- Final step after *i* iterations
  - Possible to stretch advection iterations over multiple frames
- Clipping using MD, FA or another metric
- Set depthbuffer information
- Depth-enhancing ([CCG$^+$08]) for better plasticity
- Very flexible
  - blend in colormaps

Step 4: Compositing III

- Color for each pixel with advected textures $P_i^{\lambda_1}$ and $P_i^{\lambda_2}$:

$$R = \frac{r \cdot f_{p_k^{\lambda_2}}(x,y)}{8 \cdot f^2_{p_k^{\lambda_1}}(x,y)} + e_{x,y} + light(\mathcal{L}_{x,y}),$$

$$G = \frac{(1-r) \cdot f_{p_k^{\lambda_1}}(x,y)}{8 \cdot f^2_{p_k^{\lambda_2}}(x,y)} + e_{x,y} + light(\mathcal{L}_{x,y}), \text{ and}$$

$$B = e_{x,y} + light(\mathcal{L}_{x,y}).$$

  - Silhouette texture: e and Light: $\mathcal{L}$
  - r defines ratio between both eigenvector fields in the final image

Step 4: Compositing IV

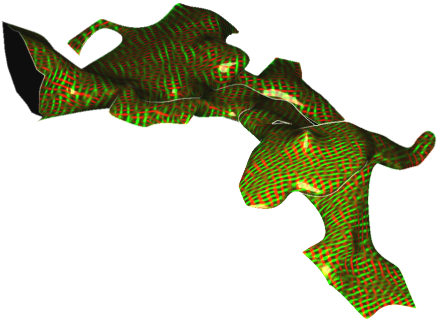
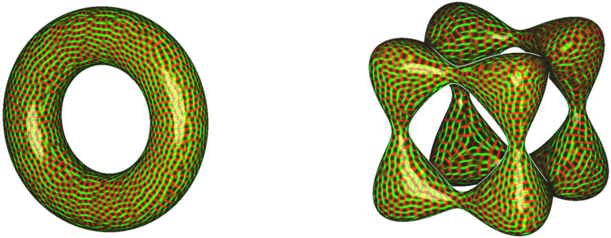


Figure: Composited image showing diffusion-directions through a fabric like structure.

Outline

1. Introduction
   - Focus
   - Tensor field visualization
   - Motivation and Goals
2. The method
   - Step 0: Input
   - Step 1: Projection to Image Space
   - Step 2: Silhouette detection
   - Step 3: Advection
   - Step 4: Compositing
3. Results
4. Problems and Further Work

Artificial Datasets



(a) Torus (b) Tangle

Figure: Implicit, C^1 steady surfaces ([KHH+07])

Small part of DTI dataset

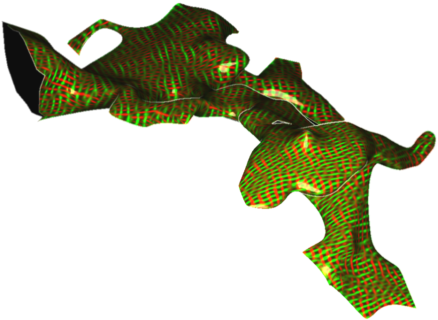


Figure: Small part of Corpus Callosum in a DTI dataset. (58624 triangles, 30 FPS (Geometry: 69%))

Neural Fibers (DTI)

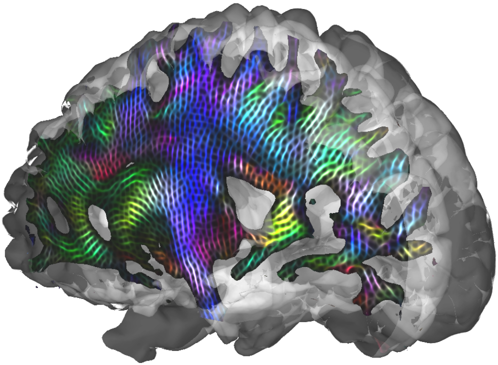


Figure: Diffusion along neural fibers (Anwander et al. [ASH+09]). (41472 Triangles, 32 FPS (Geometry: 72%))

Single Point Load (Mechanics)

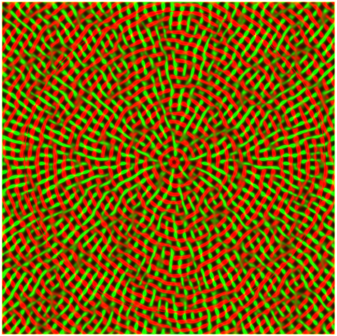


Figure: Single Point Load dataset.

Outline

1 Introduction
- Focus
- Tensor field visualization
- Motivation and Goals

2 The method
- Step 0: Input
- Step 1: Projection to Image Space
- Step 2: Silhouette detection
- Step 3: Advection
- Step 4: Compositing

3 Results

4 Problems and Further Work

Problems

- Noise texture mapping strongly dependent on quality of normals
- Minor blurring effects
- Lighting with Phong often not optimal for spatial perception of geometry

Further Work

- Reduction of rendered geometry for further performance improvement
  - Since geometry rendering is lion's share in overall rendering time
- Extend to tensor fields of higher order
- Variation of spot sizes and density in initial noise texture
  - Corresponding to eigenvalues
  - As in [HFHJ09]

Thank You for listening

Questions?

Bibliography I

- Alfred Anwander, Ralph Schurade, Mario Hlawitschka, Gerik Scheuermann, and Thomas R. Knösche.
  White matter imaging with virtual Klingler dissection.
  Human Brain Mapping 2009, 2009.

- James F. Blinn.
  Light reflection functions for simulation of clouds and dusty surfaces.
  In *SIGGRAPH '82: Proceedings of the 9th annual conference on Computer graphics and interactive techniques*, pages 21–29, New York, NY, USA, 1982. ACM.

Bibliography II

Alan Chu, Wing-Yin Chan, Jixiang Guo, Wai-Man Pang, and Pheng-Ann Heng.
Perception-aware depth cueing for illustrative vascular visualization.
In *BMEI '08: Proceedings of the 2008 International Conference on BioMedical Engineering and Informatics*, pages 341–346, Washington, DC, USA, 2008. IEEE Computer Society.

Bibliography III

- Thierry Delmarcelle and Lambertus Hesselink.
  Visualization of second order tensor fields and matrix data.
  In *VIS '92: Proceedings of the 3rd conference on Visualization '92*, pages 316–323, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press.
- Khader M. Hasan, Peter J. Basser, Dennis L. Parker, and Andrew L. Alexander.
  Analytical computation of the eigenvalues and eigenvectors in DT-MRI.
  Journal of Magnetic Resonance, 152(1):41 – 47, 2001.

Bibliography IV

- Ingrid Hotz, Z. X. Feng, Bernd Hamann, and Kenneth I. Joy.
  Tensor field visualization using a fabric-like texture on arbitrary two-dimensional surfaces.
  In Torsten Möller, Bernd Hamann, and R. D. Russel, editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*. Springer-Verlag Heidelberg, Germany, 2009.
- Yuya Iwakiri, Yuuichi Omori, and Toyohisa Kanko.
  Practical texture mapping on free-form surfaces.
  In *PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, page 97, Washington, DC, USA, 2000. IEEE Computer Society.

Bibliography V

- Aaron Knoll, Younis Hijazi, Charles Hansen, Ingo Wald, and Hans Hagen.
  Interactive ray tracing of arbitrary implicits with simd interval arithmetic.
  In *RT '07: Proceedings of the 2007 IEEE Symposium on Interactive Ray Tracing*, pages 11–18, Washington, DC, USA, 2007. IEEE Computer Society.
- G Kindlmann.
  Superquadric tensor glyphs.
  In *Proceedings of IEEE TVCG/EG Symposium on Visualization 2004*, pages 147–154, May 2004.

Bibliography VI

- James T. Kajiya and Brian P Von Herzen.
  Ray tracing volume densities.
  In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 165–174, New York, NY, USA, 1984. ACM.
- Budirijanto Purnomo, Jonathan D. Cohen, and Subodh Kumar.
  Seamless texture atlases.
  In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 65–74, New York, NY, USA, 2004. ACM.

Bibliography VII

Alan Turing.
The chemical basis of morphogenesis.
Philosophical Transactions of the Royal Society of London, 237(641):37 – 72, 1952.

Greg Turk.
Generating textures on arbitrary surfaces using reaction-diffusion.
In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 289–298, New York, NY, USA, 1991. ACM.

Bibliography VIII

Daniel Weiskopf and Thomas Ertl.
A hybrid physical/device-space approach for spatio-temporally coherent interactive texture advection on curved surfaces.
In *GI '04: Proceedings of Graphics Interface 2004*, pages 263–270, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

Bibliography IX

- David Weinstein, Gordon Kindlmann, and Eric Lundberg.
  Tensorlines: advection-diffusion based propagation through diffusion tensor fields.
  In *VIS '99: Proceedings of the conference on Visualization '99*, pages 249–253, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.
- Xiaoqiang Zheng and Alex Pang.
  Hyperlic.
  In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 33, Washington, DC, USA, 2003. IEEE Computer Society.