Reconstructing Power Cables From LIDAR Data Using Eigenvector Streamlines of the Point Distribution Tensor Field

Marcel Ritter
Institute of Basic Sciences in Civil Engineering[2]
University of Innsbruck
marcel.ritter@uibk.ac.at

Werner Benger
Center for Computation & Technology[1]
Institute for Astro- and Particle Physics[2]
werner@cct.lsu.edu
werner.benger@uibk.ac.at

ABSTRACT

Starting from the computation of a covariance matrix of neighborhoods in a point cloud, streamlines are utilized to reconstruct lines of linearly distributed points following the major Eigenvector of the matrix. This technique is similar to fiber tracking in diffusion tensor imaging (DTI), but in contrast is done mesh-free. Different weighting functions for the computation of the matrix and for the interpolation of the vector in the point cloud have been implemented and compared on artificial test cases. A dataset stemming from light detect and ranging (LIDAR) surveying served as a testbed for parameter studies where, finally, a power cable was reconstructed.

Keywords: tensor-field visualization; streamlines; mesh-free methods; particle systems; point cloud; covariance matrix; fiber tracking; LIDAR; DT-MRI

1 INTRODUCTION

Reconstructing lines from point clouds has an important application in light detection and ranging applications (LIDAR). The surveying of power lines and their geometrical analysis is of great interest for companies that transmit electrical energy. Large networks of electric facilities have to be maintained to guarantee stable electrical power supply and prevent power outages. LIDAR surveying is a suitable technique to either detect damages on the electrical facilities or detect high growing vegetation in power line corridors [19] [15]. We experiment on a new method to reconstruct linear structures, stemming from airborne LIDAR surveying. We utilize a method inspired by diffusion tensor imaging (DTI) fiber tracking developed, originally, for magnetic resonance imaging (MRI) to track neuronal structures in the human brain [5].

1.1 Related Work

Current algorithms for reconstructing power lines are usually based on data filtering followed by a segmentation of the filtered and reduced point cloud either directly on the point cloud data or on a rastered 2D image. Melzer [18] first computes a digital terrain model (DTM) by using the method by Kraus [14] to remove terrain points. The remaining points are projected onto a 2D gray-scale raster (image). A Hough-Transform (e.g. [11]) is utilized iteratively to detect straight lines. Later, Melzer [17] improved the segmentation of LIDAR data also for power cables, based on the so called mean shift clustering, originally developed for pattern recognition [9]. Liu et al. [16] introduced a methodology based on statistical analysis to first remove ground points. Then, they project points onto a 2D gray-scale raster (image) and do a Hough-Transform similar to Melzer [18], but use a different technique for the Hough-Transform [8] to detect straight lines. Jwa et al. [13] developed a four step method. First they select power-line candidates, by utilizing a voxel based Hough-Transform to recognize linear regions. After a filtering process they construct line segments based on geometric orientation rules and, finally, use a voxel-based piece-wise line detector to reconstruct the line geometries.

Weinstein et al. [23] worked on tracking linear structures in diffusion tensor data stemming from MRI. Besides following the major Eigenvector they developed some rules for overcoming areas of not linear diffusion. The flow of Eigenvectors was also used for segmentation and clustering in brain regions as, for example, shown in [6] and [20]. Jones discusses the study of connections in human brains. He states that tracking the

[1] Louisiana State University, Baton Rouge, LA-70803, USA
[2] University of Innsbruck, Technikerstraße 13/4, A-6020 Innsbruck, Austria

diffusion directions is still not solved a in stable way and is an active research area [12].

Our work is based on previous work on the direct visualization of the covariance matrix describing the local geometric properties of a neighborhood distribution within in a point cloud, the so called point distribution tensor [21].

1.2 Our Approach

In our method we do not want to remove any points but operate on the entire dataset to avoid artifacts due to a complex point removal method. Instead, we first compute the point distribution tensor for each point. Eigen-analysis of the tensor yields the major Eigenvector, which indicates the dominant orientation of a point distribution. We may follow this orientation by computing streamlines along this dominant Eigenvector field in regions where one Eigenvalue dominates, so-called linear regions. In contrast, regions where the points are distributed more isotropic, are indicated by the point distribution tensor's Eigenvalues becoming more similar values. We want to avoid these regions, as they will not correspond to power cables. This approach is very similar to the fiber-tracking approach in medical visualization, but in our case the integration of the Eigenvectors needs to be done in a mesh-free way, merely on a point distribution rather than uniform grids. Thus, it can be applied to airborne LIDAR data without resampling to uniform grids (which would reduce data resolution and introduce artifacts due to the chosen resampling method).

1.3 Overview of the Paper

Section 2 presents the mathematical background and describes the implementation of the algorithm in section 2.2. Section 2.3 shows verifications by means of simple artificial point distributions. Here, the influence of different weighting functions on the tensor computation and the vector field interpolation during streamline integration is investigated. Also, two different numerical integration schemes are tested. In section 3 one set of power cables is reconstructed from a LIDAR data set stemming from actual observations. We then explore the available parameter space for weighting and integration in order to identify the best values for the given scenario.

2 ALGORITHM

2.1 Background

In [21] we defined the "point distribution tensor" of a set of N points $\{P_i : i = 1,...,N\}$ as

$$S(P_i) = \frac{1}{N}\sum_{k=1}^{N} \omega_n(|t_{ik}, r|)(t_{ik} \otimes t_{ik}^{\tau}), \qquad (1)$$

whereby $\otimes$ denotes the tensor product, $^{\tau}$ the transpose and $t_{ik} = P_i - P_k$. $\omega_n(|t_{ik}|, r)$ is a weighting function dependent on the distance of a point sample to a center point P_i and a radius of a neighborhood r, which can be constant or defined by a scalar field on the points: $r(P_i)$. We did not find a generally optimal solution for the weighting function, but implemented seven choices for our first investigations:

$$\omega_1 = 1 \qquad (2)$$

$$\omega_2 = 1 - x/r \qquad (3)$$

$$\omega_3 = 1 - (x/r)^2 \qquad (4)$$

$$\omega_4 = r/x^2 \qquad (5)$$

$$\omega_5 = \begin{cases} 1 - \frac{3}{2}a^2 + \frac{3}{4}a^3 & 0 \le a < 1 \\ \frac{1}{4}(2-a)^3 & 1 \le a < 2 \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

$$\omega_6 = \begin{cases} (\frac{5}{2} - b)^4 - 5(\frac{3}{2} - b)^3 + 10(\frac{1}{2} - v)^b & [0, \frac{1}{2}) \\ (\frac{5}{2} - b)^4 - 5(\frac{3}{2} - b)^3 & [\frac{1}{2}, \frac{3}{2}) \\ (\frac{5}{2} - b)^4 & [\frac{3}{2}, \frac{5}{2}) \\ 0 & [\frac{5}{2}, \infty) \end{cases} \qquad (7)$$

$$\omega_7 = \begin{cases} (3-c)^5 - 6(2-c)^5 + 15(1-c)^5 & [0,1) \\ (3-c)^5 - 6(2-c)^5 & [1,2) \\ (3-c)^5 & [2,3) \\ 0 & [3,\infty) \end{cases} \qquad (8)$$

with $a := \frac{2x}{r}$, $b := \frac{2.5x}{r}$ and $c := \frac{3.0x}{r}$, illustrated in Figure 1. The three functions ω_5, ω_6 and ω_7 are typical Gauss-like spline kernel functions used in smooth particle hydrodynamics (SPH) [10]. We use the same weighting functions for interpolating the vector field during Eigenvector integration. Even though, interpolation of Eigenvectors and interpolating tensors and locally computing its Eigenvectors lead to different results, we utilize the interpolation of the Eigenvector as a simpler implementation.

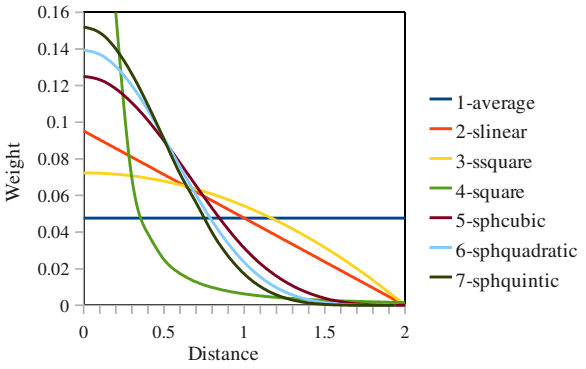


Figure 1: Different weighting functions of the distance interval 0.0 to 2.0, $r = 2.0$. Different slopes and characteristics are visualized. The *square* function (green) was clamped for axis scaling reasons and would grow further quadratically to the origin. The weights were normalized regarding to the integral of the curve in the interval. The curve numbers match the index of the weighting function: 1-average illustrates ω_1, 2-slinear illustrates ω_2, ...

We utilize tensor splats [1] for direct visualization of the tensor field. Figure 2 illustrates a point distribution along the edges of a rectangle and the corresponding tensor visualization with a neighborhood being $1/5$ of the longer rectangle edge. We then use Westin's shape

analysis method [24] to determine the so-called linear, planar and spherical shape factors. Points having a linearly distributed neighborhood are displayed as green oriented splats. Planar distributions are displayed as red disks. The linearity of the distribution tensor is shown in Figure 4 and Figure 5.

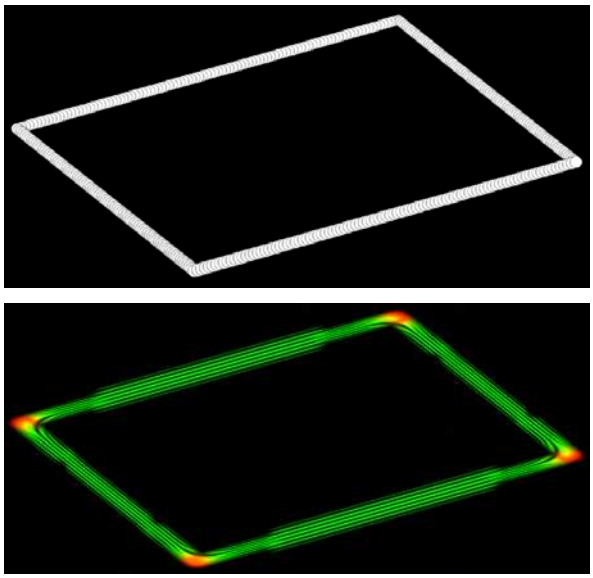


Figure 2: Distribution tensor visualization of a rectangular point distribution. *Top:* Points on a rectangle. *Bottom:* Tensor splats [1] of the point distribution tensor [21]. At each point one splat, a small textured and oriented disk, is drawn to represent the properties of the tensor's shape.

Visualizing streamlines is a common method to study vector fields. Starting from some seeding point, or initial condition, a curve $q(s)$ is computed which is always tangent to the vector field, solving the equation:

$$\dot{q}(s) = V(q(s)) \tag{9}$$

with s the curve parameter and V the vector field. Solving the differential equation at an arbitrary coordinate location Q within in the discretized data domain requires interpolation of the vector field. For mesh-free interpolation within a point cloud we use weighting functions parameterized with a specific radius of influence:

$$v(Q) = \frac{\sum_{i=1}^{N} v(P_i)\omega(|Q-P_i|,r)}{\sum_{i=1}^{N} \omega(|Q-P_i|,r)}, \tag{10}$$

with $v(P_i)$ representing the vector at point P_i.

2.2 Software Engineering Aspects

The algorithm was implemented using C++ within the VISH visualization shell [2]. The implementation extends a framework for computing integral geometries in vector fields, such as streamlines, pathlines or time surfaces. The streamline integration and visualization is separated into three different components: seeding, integration and displaying. The first component defines the initial conditions or seeding geometry. For computing streamlines within vector fields seeding points are sufficient. However, for streamlines of Eigenvector fields also an initial direction must be specified, because the Eigenvector is undirected. Integration based on an orientation continuing an user-chosen direction must be possible. Thus, requiring also a vector field on initial seeding points to disambiguate the Eigenvectors' orientations into unique directions.

Two new integration modules were developed. The first one extends the original streamline module, which was designed for vector field integration in uniform and curvilinear multi-block grids [4], to Eigenvector field integration. The second module expands this method further to allow integrating Eigenvector fields on mesh-free grids. One of the seven weighting functions (Equations 2, 3, 4, 5, 6, 7 and 8) and the radial influence weighting parameter can be specified for the interpolation of the Eigenvector inside the field domain. A range query on a KD-tree returns the points and their distances within the neighborhood of radius r. Equation 10 is utilized and Eigenvectors are aligned in orientation with respect to the Eigenvector of the closest neighbor. The Eigenvector is reversed if the dot product is negative. The integration of the streamline stops when the neighborhood becomes empty. Both integration modules support two different numeric schemes for the integration: explicit Euler and DOP853 [7]. Explicit Euler is used to get a fast yet inaccurate result. DOP853 is more expensive due to its adaptive stepsize but gives highly accurate results. When aiming at the same accuracy, DOP853 is faster than the Euler method by orders of magnitude. It is a Runge Kutta method of order eight using order five and three for error estimation and adaptive step size control, providing dense output. Accuracy measures and timing measures comparing the two integration methods were done, e.g., in [3].

The display module utilized here is reused from earlier development and implements color-coded illuminated lines utilizing OpenGL, allowing interactive navigation through the generated streamlines. Other modules, such as displaying ribbons [3] are also available.

2.3 Test Cases

We investigate the two Eigenvector integration modules on an uniform grid and on mesh-free grids. The Eigenvector field of a DTI-MRI scan [1], originally given on a uniform grid (128x128x56), was converted into a mesh-free grid, a point cloud holding the same Eigenvectors: Figure 3 (a) shows a volume rendering of the trace of the diffusion tensor along with the streamlines, revealing some brain structure and the location of the streamlines. Figure 3 (b) shows the comparison of Eigenvector streamlines computed on the uniform grid (blue) and Eigenvector streamlines computed in the point cloud (white). Both integrations were done with explicit Euler and a step size of 0.05. The size of a uniform grid cell is about 0.2, thus, utilizing about four integra-

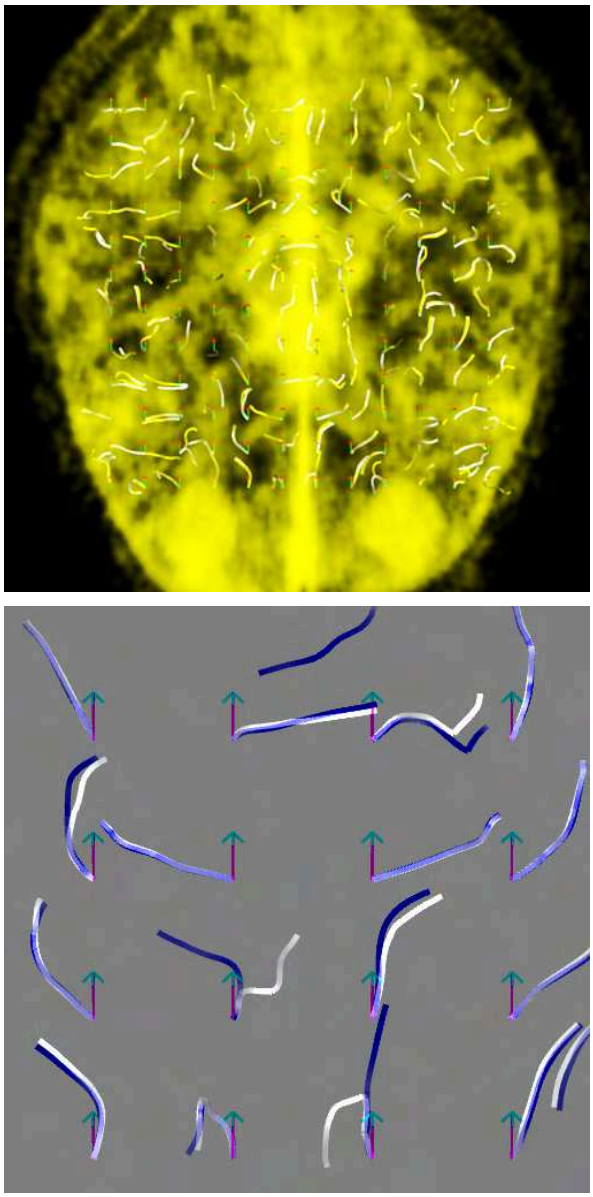
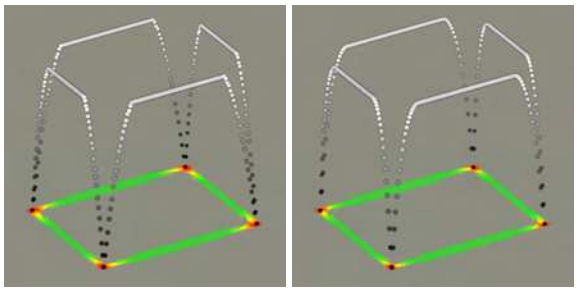
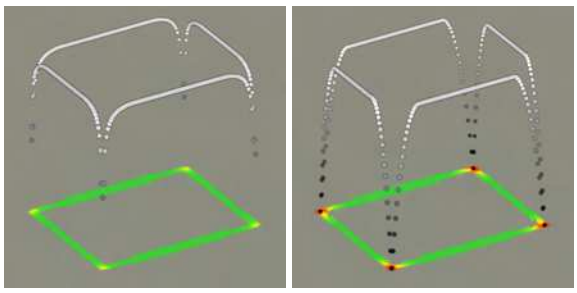


Figure 3: Comparison of the influence of integration of an Eigenvector field given on an uniform and a mesh-free grid. A mesh-free grid was generated from the uniform for testing. The arrows mark the start positions and directions of small Eigenstreamlines of a MRI diffusion tensor field. Streamlines on the uniform grid are blue. On the mesh-free grid they are white.

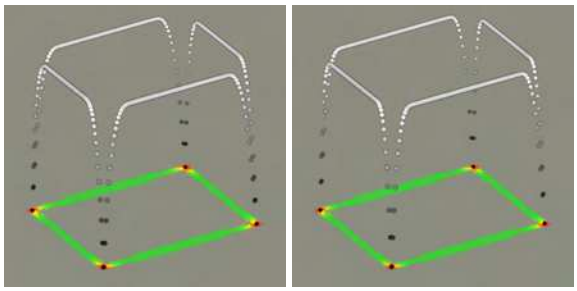
tion points per grid cell and requiring data interpolation within each cell. The length of each streamline is set to 1.0. Tri-linear interpolation was chosen for the uniform grid to compare the results with the linear weighting function ω_2 (*slinear*) for the mesh-free grid. The generated lines coincide on most cases. About 9% (13 of 144) do not coincide well. Some start in different directions. Here, the seeding vector field is almost perpendicular to the initial direction and the influence of the interpolation method results in different initial streamline directions. This issue could be cured by integrating Eigenvector streamlines in both directions starting from the initial seeding points, which would also allow avoiding the seeding vector field.



(a) average, slinear



(b) square, sphcubic

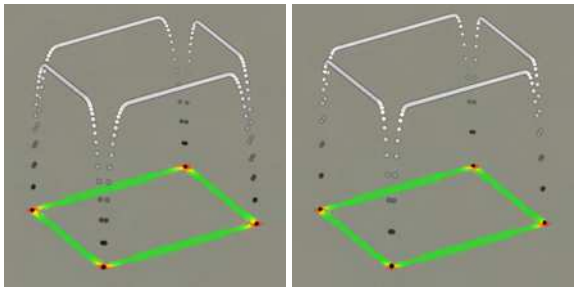


(c) sphquadratic, sphquintic

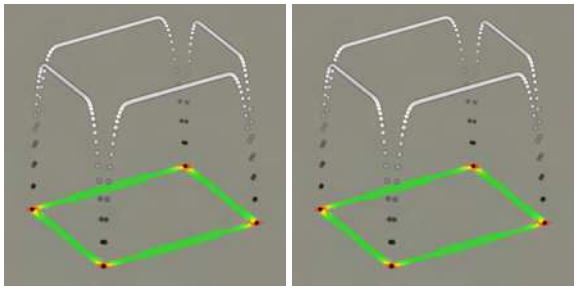
Figure 4: Influence of different weighting functions on the scalar field *linearity*, compare Figure 1. The linearity is illustrated by offset and over-scaling in z-axis, and gray-scale color-map on the points. Tensor splats directly show the distribution tensor.

Next, the influence of the different weighting functions on the computation of the distribution tensor was investigated. We define an analytic distribution of points along a rectangle as test case for computing the point distribution tensor. The rectangle is set up using a width of 10 and a height of 8. The radius parameter for the neighborhood is $r = 0.2$. Figure 4 illustrates the point distribution tensor using tensor splats and its corresponding linear shape factor by offsetting, over-scaling and a gray-scale color-map. The offsetting approach for the linear shape factor clearly illustrates the influence of the weighting: The "*average*" method resulting in a very abrupt change in the slope around corners points. The "*slinear*" weighting function results in smoother changes and a more localized influence, since closer points are weighted stronger than more distant points. *Square* shows the smoothest result. The three SPH spline kernels have an increasing locality with higher order of the kernel, when comparing *sphcubic*, *sphquadratic* and *sphquintic*. This is demonstrated in Figure 5 as well: Figure 5(a) shows the result of the

cubic and quadratic SPH kernel function. When the radius of the neighborhood is increased to match the kernels there is no visible difference between the *sphcubic* in Figure 5(a), *sphquadratic* and *sphquintic* in Figure 5(b) in the resulting *linearity*.



(a) *sphcubic* $r = 0.2$, *sphquadratic* $r = 0.2$



(b) *sphquadratic* $r = 0.25$, *sphquintic* $r = 0.3$

Figure 5: Different orders of the SPH kernel functions are compared, see Figure 1. *(a) sphcubic* and *shpquintic* using the same radius for the neighborhood. *(b) sphquadratic* and *shpquintic*, with adjusted neighborhood radius, have a similar result as the *sphcubic* (a)-left.

The influence of the integration scheme on the Eigenstreamline integration is demonstrated in Figure 6. The distribution tensor of a circular point distribution was computed using the *ssquare* weighting function. Tensor splats show the undirected Eigenvector, vector arrows show how the vector is directed within the internal vector representation. One Eigenstreamline is seeded downwards at the rightmost point of the circular point distribution and follows the undirected vectors. The top image shows Euler integration. Decreasing the step size would result in a more accurate integration. But, closing the gap of the integrated circle requires such a small step size, that the Runge Kutta method outperforms the Euler method. The 8^{th} order Runge Kutta method successfully closes the gap and reconstructs a circle from the circular point distribution, as shown in the bottom image. Also, a square-shaped point distribution was tested as shown in Figure 7. The length of a side is 10. Here, the influence of different weighting functions on the interpolation of the Eigenvector field was investigated. The distribution tensor was computed using the *ssquare* weighting function with $r = 2$. An Eigenstreamline is seeded downwards in the mid of the right edge. It follows the undirected vectors and flows around the corners of the rectangle. At each corner some error is introduced and the streamline is moving apart from

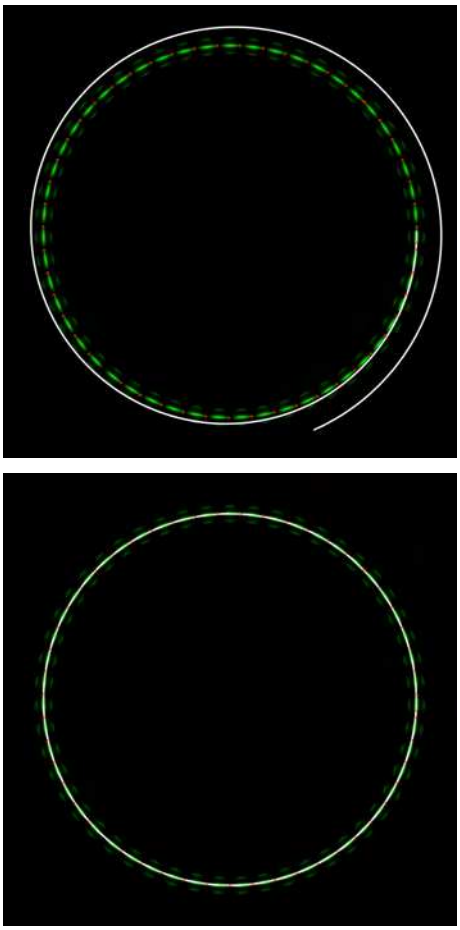


Figure 6: Comparison of different numerical integration schemes in a circular point distribution. One streamline (white) is seeded at the east-pole of the circle pointing southwards. Tensor splats and vector arrows illustrate the point distribution tensor and major Eigenvector. Note, that the Eigenvectors change orientation at north-east and south-west. *Top:* explicit Euler. *Bottom:* DOP853.

the original point distribution. Integration was done using the DOP853 method. Different weighting functions, mostly with $r = 1$, were tested for vector field interpolation. The length of the horizontal gap between the end and the start of the streamline was used as a measure for the integration error. Figure 8 shows the different errors in a bar diagram. The two best results were achieved using the *ssquare* and *average* weighting function.

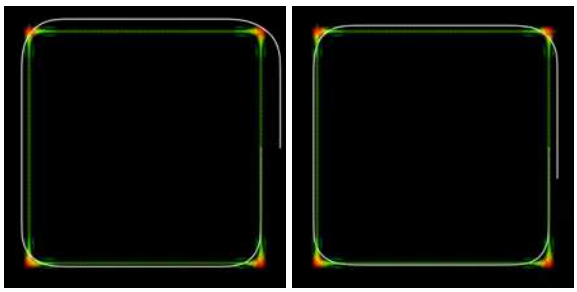


Figure 7: Comparison of Euler and DOP853 streamline integration on a square-shaped point distribution. Tensor splats and Eigenvectors are visualized besides the streamline (white) seeded downwards at the center of the right edge of the rectangle.

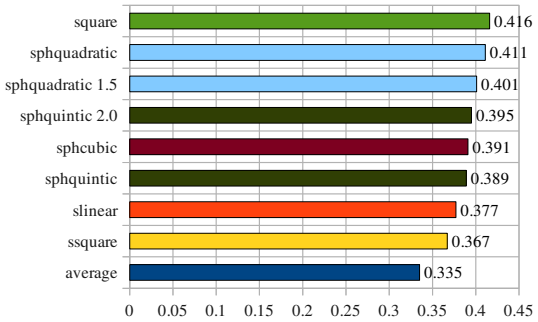


Figure 8: Comparison of errors in the square integration using different weighting functions for the vector interpolation. The weighting function for computing the tensor was *ssquare*, compare Figure 1. The values represent the horizontal distance between start and end point of the streamlines. The square's length is 10.0. The colors of the bars match the colors in Figure 1.

3 RESULTS

We used a dataset with circa eight million points covering a water basin close to the Danube in Austria. It was acquired by a Riegl's hydrographic laser scanner VQ-820G [22]. Figure 9 shows the point cloud colored by the *linearity* of a distribution tensor analysis. Here, we wanted to extract one power cable. The cable in the mid of the three lowest power cables suspended from the tall power pole. The white arrows mark the explicitly user-specified position and direction used as initial conditions of the streamline integration.

Different parameters and combinations of weighting functions for the tensor computation and the Eigenvector interpolation were investigated. The choice of a certain neighborhood radius and good weighting functions was crucial to successfully follow the 280 m long power cable. 41 parameter combinations were tested. For the tensor computation different radii $r = 0.5, r = 1.0$ and $r = 2.0$ and the weighting functions *average*, *slinear*, *ssquare* and the SPH kernels for the tensor were used. For the vector interpolation radii $r = 0.25, r = 0.5, r = 1.0, r = 2.0$ and $r = 3.0$ and all seven weighting functions were used. Figure 10 shows a view along the power cable, with a non optimal configuration. The Eigenstreamline is not following the cable to the end because it moves apart more than 1.0 m from the cable, resulting in an empty neighborhood during integration.

Best results were achieved by using the *ssquare* weighting with $r = 2.0$ for tensor computation and the *sphquintic* weighting with $r = 3.0$ for the vector interpolation. Results show that a more smooth weighting in the tensor computation and a more local interpolation weight are a good combination for reconstructing linear structures. Using the same weighting for tensor computation and vector interpolation did not work, see Figure 11 (b). The global error of the reconstruction at the end of the power cable is about 80 cm and needs to be further improved. The cable could only be followed using DOP853 integration. Explicit Euler failed to produce acceptable results. When comparing Figures $11(a)$ and $11(c)$ the global error is almost the same. The main difference is the local shape of the Eigenstreamline. A larger vector interpolation radius results in a smoother curve. Figure $11(c)$ shows the best reconstruction of the investigated technique and described parameters.

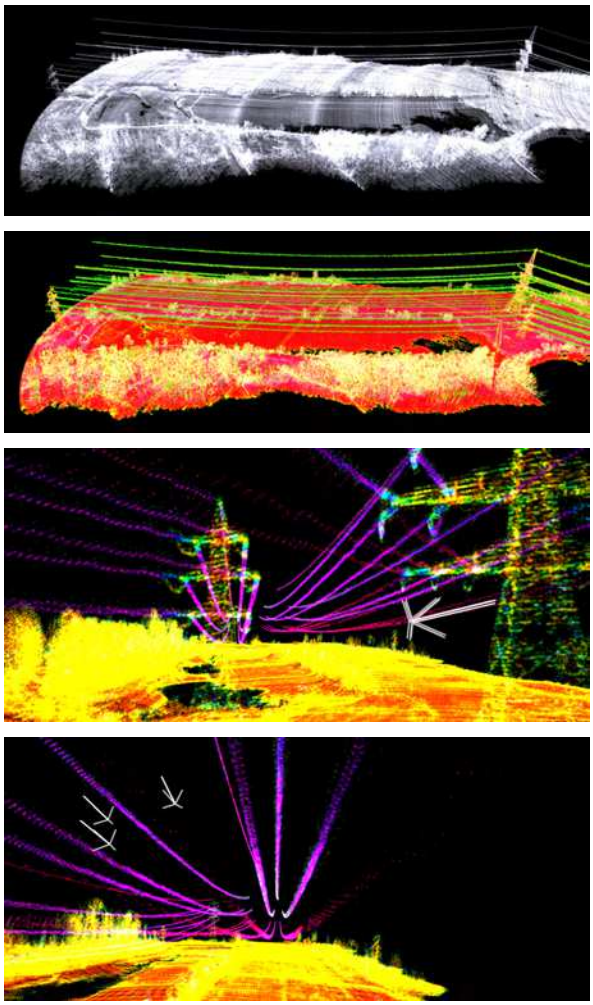


Figure 9: Overview of the LIDAR data set. The two upper images show the point cloud as points and as tensor splats (taken from [21]). In the two lower images points are colored by *linearity*. Three arrows mark the explicitly user-specified seeding points and directions of the streamline computation located at the mid lower power cable (magenta) of the larger power pole.

4 CONCLUSION

A new method of reconstructing power cables, or other linear structures in general, in point clouds was presented. The method employs the point distribution tensor as presented in previous work [21]. Different weighting functions for the tensor computation and the interpolation of the major Eigenvector field were implemented and compared. Streamline integration was verified on artificial test cases and applied to a LIDAR point cloud dataset acquired from actual observations. Finally, a power cable was reconstructed and visualized using this dataset.

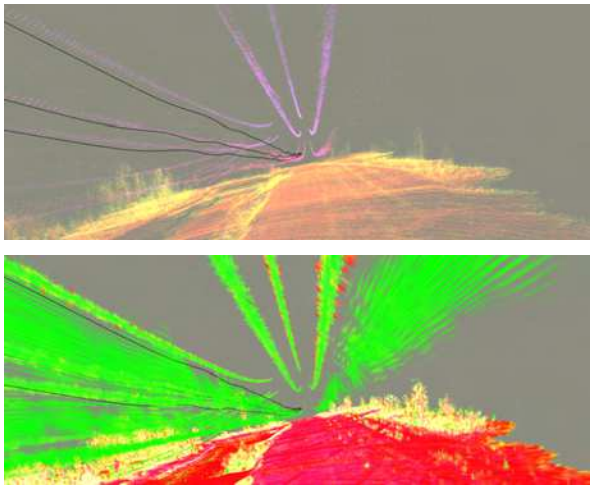
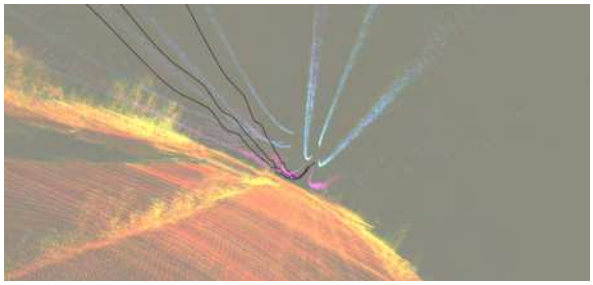
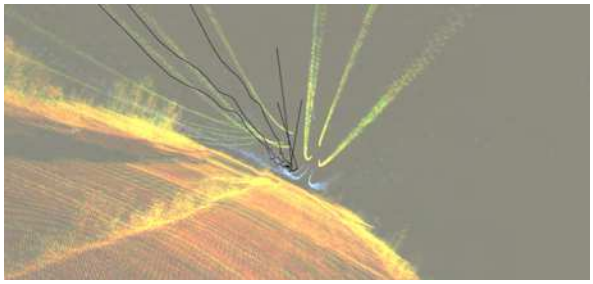


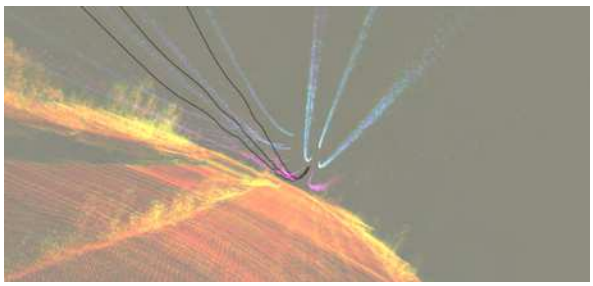
Figure 10: Power cable reconstruction via streamlines. The distribution tensor was computed using the *average* $r = 2.0$ weighting and the vector interpolation was done with the *ssquare* $r = 1.0$ weighting. *Top:* Points colored by *linearity*. *Bottom:* Tensor splats illustrate the distribution tensor. Streamlines are moving apart from the power cable and break before they can reconstruct the full 280 m of cable.



(a) Tensor: *ssquare r=2*, Vectorfield: *sphquintic r=1*



(b) Tensor: *sphquintic r=2*, Vectorfield: *sphquintic r=2*



(c) Tensor: *ssquare r=2*, Vectorfield: *sphquintic r=3*

Figure 11: Comparison of different parameters and weighting function combinations of the computation, finally resulted in a successfully reconstructed power cable. The LIDAR point cloud is colored by *linearity* of the distribution tensor. The three Eigenvector streamlines reconstruct a 280 m long cable.

5 FUTURE WORK

Other weighting functions for computing the tensor and doing the interpolation during the streamline integration need to be tested. Automatic determination of the optimal combination of weighting functions and also their parameters will be the goal of further investigations. Seeding points and directions for computing the streamlines need also to be chosen automatically, for example, by taking tensor properties into account. Following the major Eigenvector of points with high *planarity* or *sphericity* needs to be prevented during streamline integration. Finally, more datasets should be explored to stabilize the method. Furthermore, minor changes of the algorithm would enable streamline integration in datasets stemming from SPH simulations.

ACKNOWLEDGMENT

Many thanks to Frank Steinbacher for proving the LIDAR data. This work was supported by the Austrian Science Foundation FWF DK+ project *Computational Interdisciplinary Modeling* (W1227) and grant P19300. This research employed resources of the Center for Computation and Technology at Louisiana State University, which is supported by funding from the Louisiana legislature's Information Technology Initiative. This work was supported by the Austrian Ministry of Science BMWF as part of the UniInfrastrukturprogramm of the Forschungsplattform Scientific Computing at LFU Innsbruck.

REFERENCES

[1] W. Benger, H. Bartsch, H.-C. Hege, H. Kitzler, A. Shumilina, and A. Werner. Visualizing Neuronal Structures in the Human Brain via Diffusion Tensor MRI. *International Journal of Neuroscience*, 116(4):pp. 461–514, 2006.

[2] W. Benger, G. Ritter, and R. Heinzl. The Concepts of VISH. In 4^{th} *High-End Visualization Workshop, Obergurgl, Tyrol, Austria, June 18-21, 2007*, pages 26–39. Berlin, Lehmanns Media-LOB.de, 2007.

[3] W. Benger and M. Ritter. Using geometric algebra for visualizing integral curves. In E. M. S. Hitzer and V. Skala, editors, *GraVisMa 2010 - Computer Graphics, Vision and Mathematics for Scientific Computing*. Union Agency - Science Press, 2010.

[4] W. Benger, M. Ritter, S. Acharya, S. Roy, and F. Jijao. Fiberbundle-based visualization of a stir tank fluid. In 17^{th} *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 117–124, 2009.

[5] T. E. Conturo, N. F. Lori, T. S. Cull, E. Akbudak, A. Z. Snyder, J. S. Shimony, R. C. Mckinstry, H. Burton, and M. E. Raichle. Tracking neuronal

fiber pathways in the living human brain. *Proc Natl Acad Sci U S A*, 96(18):10422–10427, Aug. 1999.

[6] M. Descoteaux, L. Collins, and K. Siddiqi. A multi-scale geometric flow for segmenting vasculature in mri. In *In Medical Imaging Computing and Computer-Assisted Intervention*, pages 500–507, 2004.

[7] S. N. E. Hairer and G. Wanner. *Solving ordinary differential equations I, nonstiff problems, 2nd edition*. Springer Series in Computational Mathematics, Springer-Verlag, 1993.

[8] L. A. Fernandes and M. M. Oliveira. Real-time line detection through an improved hough transform voting scheme. *Pattern Recognition*, 41(1):299 – 314, 2008.

[9] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32 – 40, jan 1975.

[10] D. A. Fulk and D. W. Quinn. An analysis of 1D smoothed particle hydrodynamics kernels. *Journal of Computational Physics*, 126(1):165–180, 1996.

[11] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.

[12] D. K. Jones. Studying connections in the living human brain with diffusion mri. *Cortex*, 44(8):936 – 952, 2008.

[13] Y. Jwa, G. Sohn, and H. B. Kim. Automatic 3d powerline reconstruction using airborne lidar data. *IAPRS*, XXXVIII(2004):105–110, 2009.

[14] K. Kraus and N. Pfeifer. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 53(4):193 – 203, 1998.

[15] Z. Li, R. Walker, R. Hayward, and L. Mejias. Advances in vegetation management for power line corridor monitoring using aerial remote sensing techniques. In *Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on*, pages 1 –6, oct. 2010.

[16] Y. Liu, Z. Li, R. F. Hayward, R. A. Walker, and H. Jin. Classification of airborne lidar intensity data using statistical analysis and hough transform with application to power line corridors. In *Digital Image Computing : Techniques and Applications Conference (DICTA 2009)*, Melbourne, Victoria, December 2009. IEEE Computer Society.

[17] T. Melzer. Non-parametric segmentation of als point clouds using mean shift. *Journal of Applied Geodesy*, 1(3):159–170, 2007.

[18] T. Melzer and C. Briese. Extraction and modeling of power lines from als point clouds. In *Proceedings of 28th Workshop*, pages 47–54. Österreichische Computer Gesellschaft, 2004. talk: Austrian Association for Pattern Recognition (ÖAGM), Hagenberg; 2004-06-17 – 2004-06-18.

[19] S. Mills, M. Gerardo, Z. Li, J. Cai, R. F. Hayward, L. Mejias, and R. A. Walker. Evaluation of aerial remote sensing techniques for vegetation management in power line corridors. *IEEE Transactions on Geoscience and Remote Sensing*, October 2009.

[20] M. Persson, J. Solem, K. Markenroth, J. Svensson, and A. Heyden. Phase contrast mri segmentation using velocity and intensity. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 119–130. Springer Berlin - Heidelberg, 2005.

[21] M. Ritter, W. Benger, B. Cosenza, K. Pullman, H. Moritsch, and W. Leimer. Visual data mining using the point distribution tensor. In *IARIS Workshop on Computer Vision and Computer Graphics - VisGra 2012*, Feb-Mar 2012.

[22] F. Steinbacher, M. Pfennigbauer, A. Ulrich, and M. Aufleger. Vermessung der Gewassersohle - aus der Luft - durch das Wasser. In *Wasserbau in Bewegung ... von der Statik zur Dynamik. Beitrage zum 15. Gemeinschaftssymposium der Wasserbau Institute TU München, TU Graz und ETH Zürich*, 2010.

[23] D. Weinstein, G. Kindlmann, and E. Lundberg. Tensorlines: advection-diffusion based propagation through diffusion tensor fields. In *Proceedings of the conference on Visualization '99: celebrating ten years*, VIS '99, pages 249–253, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[24] C. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. Jolesz. Geometrical diffusion measures for mri from tensor basis analysis. In *Proceedings of ISMRM, Fifth Meeting, Vancouver, Canada*, page 1742, Apr. 1997.