Eurographics Conference on Visualization (EuroVis) (2013)
M. Hlawitschka and T. Weinkauf (Editors)

Short Papers

Trajectory Density Projection for Vector Field Visualization

A. Kuhn[1], N. Lindow[1], T. Günther[2], A. Wiebel[3], H. Theisel[2], H.-C. Hege[1]

[1] Zuse-Institute Berlin (ZIB), [2] University of Magdeburg, [3] Hochschule Coburg, Germany

Abstract

Trajectory visualization is an important tool to capture intrinsic characteristics of vector fields. However, often this class of geometric visualization suffers from visual clutter due to the number of curves that occlude each other and may cover relevant features. Attempts trying to avoid this effect often require complex pre/post-processing for seeding, clustering, or filtering of relevant lines. In this work we present a simple, yet effective technique for rendering large amounts of trajectories, which highlights features of their projected density. The technique exploits capabilities of modern graphics hardware and avoids explicit feature extraction. We propose efficient schemes for its computation, provide suggestions for sensible parameter setups, and discuss important implementation aspects. In addition, we demonstrate the effectiveness of our approach in several application scenarios.

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

1. Introduction

Rendering of large sets of lines is a relevant topic in a variety of visualization fields. Especially for the visualization of integral curves in vector fields, numerous techniques to improve their interactive visual presentation [MLP*10] have been developed. Nevertheless, compactly visualizing internal structures of such curve sets remains a challenging problem. In general, solving it requires sophisticated pre-processing and/or post-processing techniques to reduce occlusion of relevant structures and to avoid visual clutter. In this paper, we present a fast, direct rendering approach, which projects the density of line sets along the view direction on the screen. For this purpose, we present a modified version of the method by Merhof et al. [MSE*06a], which was originally developed in the context of brain fiber visualization. Our method is furthermore inspired by line density visualizations for which Park et al. [PYH*06] have shown that they are suitable to highlight topologically relevant structures. To provide fast, intuitive and appealing visualizations of dense integral curve sets emphasizing structural characteristics of the vector field, we combine both methods and extend them toward high dynamic range rendering (HDR). We demonstrate the utility of this approach to identify features such as regions of compression, expansion, attracting and repelling points, as well as closed stream lines. Due to the direct processing, we avoid intermediate discretization and explicit extraction of topological structures. This makes our approach more stable to compute, easier to use, and allows for higher resolution accuracy, compared to previous density-based approaches [PYH*06]. The main contribution of this work is a density projection of sets of trajectories that is interactive and memory-friendly. The technique allows for direct 3D visualization of topological (asymptotic) features without an explicit extraction and operates on straight-forward uniform seeding as input. We demonstrate our method using several examples, including potential fields around molecules, time-dependent magnetic fields, and oceanic flow simulations.

2. Related Work

The use of integration-based geometry to capture important features within vector fields has a long tradition in data visualization [LHD*04, MLP*10, PPF*11]. The visual presentation of integral curves was addressed by a number of rendering approaches, improving on the perception of orientation [ZSH96, MPSS05], flow properties [SGS05], depth [EBRI09], and spatial proximity [EHS13]. Yet, all these line rendering techniques require a well-conceived placement (i.e., seeding) of lines to trade the representation of vector field properties for the occlusion and clutter, introduced by too many lines. This maximization of perceivable information was approached in different ways, i.e., by sophisticated seed placement [ATR*08, SLCZ09, XLS10, YWSC12, MJL*12], line selection from precom-

DOI: 10.2312/PE.EuroVisShort.EuroVisShort2013.031-035

puted line sets [MCHM10, GBWT11] or viewpoint selection [LMSC11, TMWS13]. Park et al. [PYH*06] used the density of trajectories to accentuate topological structures in steady vector fields, by mapping the sampled density via transfer functions to color and opacity. In 3D, the achievable quality and accuracy relates to the resolution and coincidentally the consumed memory of the discrete density volume. Wiebel et al. [WCW*11] proposed to use the time-dependent density of particles to find attraction points in unsteady flow fields. Trajectory density was used to increase spatial resolution in diffusion-weighted imagery by utilizing intra-voxel information [CTJC10]. Related approaches, avoiding explicit feature extraction, were applied to solar dynamics data, using volume visualizations of 3D LIC fields [MSM*12] and to geo-spatial scatter plot data, using kernel density estimations [LH11]. More generally, composite density maps for 2D trajectories have been shown to be a suitable visualization tool by Scheepens et al. [SWvdW*11]. The relation of density features in continuous line visualizations of higher-dimensional parameter spaces has been examined by Lehmann and Theisel [LT11]. Besides, high-performance geometry and particle blending techniques are frequently used for visual effects (e.g., fractal rendering or games [PF05, Lor05]).

3. Method

Our method creates a view-dependent density projection of a set of trajectories, given as polylines. This is done in a single rendering pass without prior density field sampling [PYH*06] or 3D LIC advection [MSM*12]. Instead, we create a triangle strip with an initial width set to zero for each polyline. In the vertex shader, we align the strip to the camera and expand it to a predefined line width, similar to Merhof et al. [MSE*06b]. The density projection is realized by an additive or subtractive color blending scheme, which emphasizes areas of increased trajectory density. To steer the projection intensity and the visual smoothness, we use a fall-off function that reduces the opacity at the line strip boundaries. In the following, we describe the rendering pipeline using OpenGL (see also Fig.1).

3.1. Rendering Pipeline

For every point of each polyline, we create two vertices. Each vertex stores the point position $\mathbf{p}_i$, an approximation of the tangent vector $\mathbf{t}_i$, a radius value r_i and a single value for a color lookup. Thus each vertex consists of two 4-dimensional vertex attributes. Notice that we use $-r_i$ for the first vertex of each line point. The vertices are stored in a vertex buffer object and each line is rendered as a triangle strip. In the vertex shader we compute the expansion of the triangle strip according to the camera. For this, the expansion vector $\mathbf{e}_i$ of $\mathbf{p}_i$ is given by $\mathbf{e}_i = (\mathbf{c}-\mathbf{p}_i)\times\mathbf{t}_i$, where $\mathbf{c}$ is the camera position (see Fig. 1). Let $\tilde{\mathbf{e}}_i$ be the normalized vector of $\mathbf{e}_i$. The final vertex position is $PM(\mathbf{p}_i + r_s r_i \tilde{\mathbf{e}}_i)$, where PM is the modelview projection matrix and r_s is a global scale for the line width. Finally, we pass for each vertex the sign of r_i as 1 or -1 and the value for the color lookup to the fragment shader. In the fragment shader, we fetch the color by a color map texture lookup. For subtractive blending (see below), we invert the color by one minus the color channel for each channel. Furthermore, we compute an alpha blending value α to fade out lines toward the triangle strip boundary. Let s be the sign value, which ranges from -1 to 1 along the strip width. First we continuously map s to 1 in the middle of the strip and to 0 at the boundaries. Afterwards, we use a gamma correction function to compute α depending on s. Thus, α is given by $(1-|s|)^{e_f}$, where e_f is a user-defined fall-off parameter, which controls the visual smoothness of the triangle strip across its boundaries.

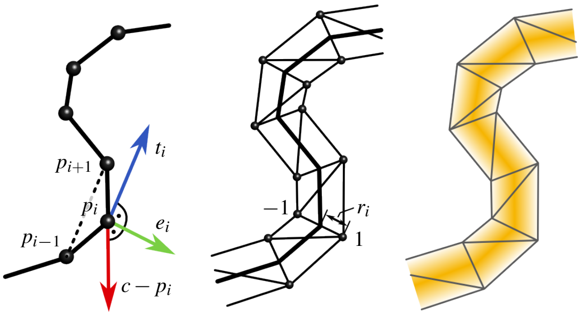


Figure 1: *Illustration of the rendering pipeline. Computation of the tangent vector t_i and the expansion vector e_i (left), expansion of the triangle strip (middle), and shading with a fall-off function (right).*

We implemented our technique for an additive and subtractive blending mode. The additive mode uses the blending equation `GL_FUNC_ADD`, which adds up colors and creates the impression of glowing lines. The subtractive blending mode is realized by `GL_FUNC_REVERSE_SUBTRACT`, which subtracts new colors from the background and creates the impression of ink painting. The correct blending factors are given by `glBlendFunc(GL_SRC_ALPHA, GL_ONE)`. Since both modes are independent of the depth order, we can render pure lines in a single pass without depth sorting and depth test. In combination with opaque geometry, we enable the depth test, render the opaque objects first and disable the writing of depth values for the lines by `glDepthMask(false)`. Note that this pipeline can be modified using geometry shaders to create the triangle strips on-the-fly from the polylines. The described algorithm can be split into multiple phases if the line set does not fit into graphics memory at once. This requires to alternate between creating new geometry (e.g., by field line integration), rendering of the results into a global buffer, and storing new seed points while removing the previous geometry. In order to avoid large bright or dark regions when using one of the blending modes, we apply high dynamic range rendering with a simple tone mapping function.

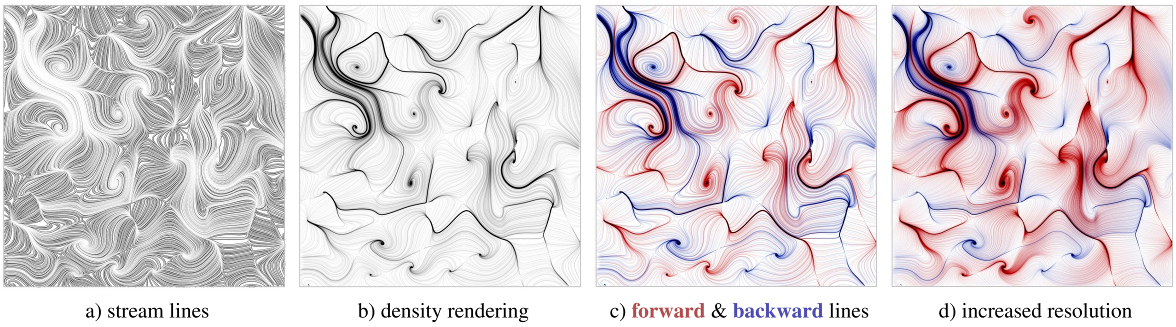


Figure 2: ***Random 2D vector field:*** *Density projections visually enhance asymptotic topological features without prior extraction [PYH*06]. Figure a) shows the original trajectories seeded on a regular* 48×48 *grid, b) the blended triangle strips* ($r_s = 0.01, e_f = 5$)*) c) shows separated* ***forward (red)*** *and* ***backward (blue)*** *trajectories, d) on a* 128×128 *grid.*

This function maps color values component-wise from $[0,\infty)$ to $[0,1]$ by $1 - e^{e_t \cdot v}$, where v is a channel of the color and e_t is a negative parameter steering the strength of the mapping. For the subtractive blending, we have to apply a pre-mapping, because the color channels lie in the range of $(-\infty, 1]$. Finally, a gamma correction is necessary to avoid color changes by the tone mapping. In summary, our rendering contains four interactive parameters: A global line width scale r_s, an alpha blending fall-off parameter e_f, a tone mappign parameter e_t, and e_g for the gamma correction. Depending on r_s and e_f both parameters adjust the resulting brightness and contrast.

3.2. Application to Vector Fields

In the context of vector fields, trajectories (e.g., stream lines, path lines, see [WHT12]) are computed by integrating a given seed geometry within the field. In order to largely eliminate the influence of seeding and integration time, we suggest to use dense uniform seeding in the complete domain and maximum integration length (i.e., until the domain boundary, stagnation points, or a maximum number of integration steps are reached). Suitable parameter setups for our method generally depend on the visualization goal and underlying field properties. Starting from the initial settings, more detailed visualizations of specific features of interest can be obtained by interactive parameter adjustment. Rendering the final blended color value emphasizes vector field features that influence the local field line density. In static fields with non-zero divergence, this is especially true for asymptotic features, such as critical points (i.e., attracting or repelling foci, sinks or sources [PYH*06]) and separatrices. Special features are closed orbits, on which trajectories repeatedly trace the same curve with growing number of integration steps. Note that especially in 3D not all types of (topological) features necessarily influence the density of nearby trajectories. Hence, density visualizations are especially useful for primary interactive visual inspection, but cannot replace a numerical extraction of features (e.g., see Weinkauf et al. [WST*07]).

4. Results

In this section, we provide a set of selected examples to demonstrate the usefulness of our approach in different scenarios. If not stated differently, we used an adaptive Runge-Kutta integrator RK4(3) with $h_{min} = 0.001, h_{max} = 0.05$ and a dense seeding across the whole domain.

Analytic Examples To illustrate the correspondence between the projected density and topological features, Figure 2 shows a steady 2D random field and Fig. 3 the analytic approximation of a 3D potential field of a benzene molecule [ZSH96]. Figure 3 a) shows the original trajectories integrated in **forward** and **backward** direction. In Figure 3 b) critical points have been extracted using the framework described by Weinkauf et al. [WST*07] and are shown together with our trajectory density projection. In contrast to the 2D case (Figure 2), saddle points are not visible in the projection rendering, while orientation and spread of *inflow* sectors around sinks and sources are emphasized.

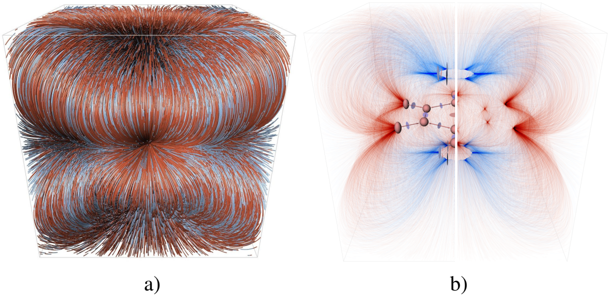


Figure 3: *Density rendering of the (symmetric) benzene data set, Figure a) shows* 30.000 *randomly seeded stream lines, b) on the left side: density with critical points [WST*07], on the right side: solely the density rendering* ($r_s = 0.01, e_f = 12$) *separated into* ***forward*** *and* ***backward*** *curves.*

Magnetic Field This example shows a series of simulated magnetic fields that depict the progress of magnetic flux decay over time. The magnetic field has a resolution of 256^3

and is given on a uniform regular grid structure. The data set shown in Figure 4 illustrates a trefoil knot configuration that collapses over time. For our visualization, we used 10.000 instantaneous field lines with constant random seeds for each time step. Despite smaller variations, the density projections highlight stable structures over the considered time interval in a frame-coherent manner (see accompanying video).

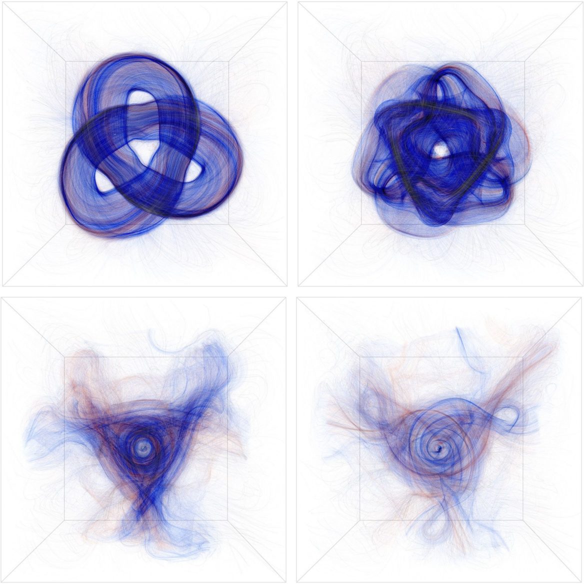
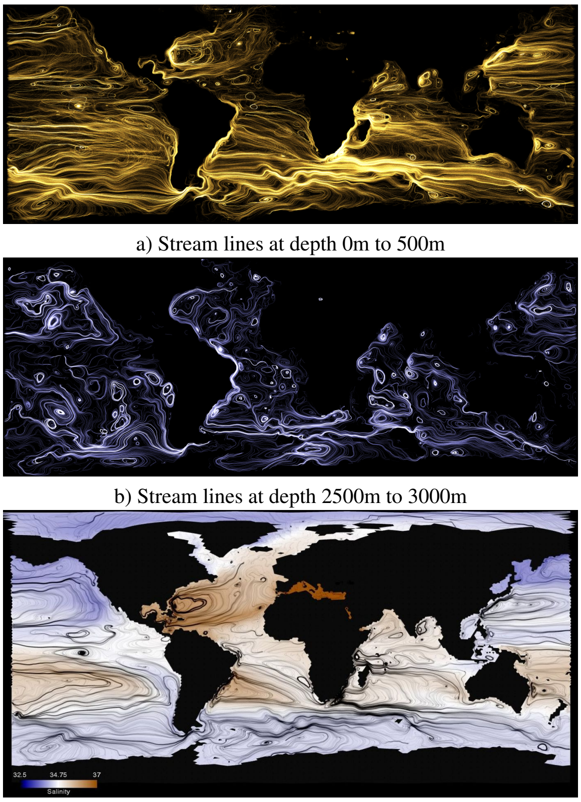


Figure 4: ***Trefoil dataset:*** *Density of* 10.000 *stream lines, randomly seeded, separated in* ***forward*** *and* ***backward*** *direction with constant line strip parameters* $r_s = 0.04$ *and* $e_f = 30$ *for the time steps 20, 50, 130, and 200.*

Ocean Dataset The final example shows the results for a oceanic flow from a climate model simulation. The simulation has been performed on a icosahedral grid, covering the complete Earth with a total amount of 1.7 Million nodes and 3.2 Million cells. Each cell contains flow velocity information, as well as additional physical quantities, such as salinity and temperature of the water. Density projection rendering allows to obtain a comparative view across dense trajectory sets, seeded at different depth levels of the data set. Figure 5 shows a uniform seeding on a $90 \times 45 \times 50$ grid in the depth range of a) 0m to 500m and b) 2500 to 3000m. Our method emphasizes closed currents and dominant streams induced by the velocity field as illustrated in Fig. 5c).



a) Stream lines at depth 0m to 500m

b) Stream lines at depth 2500m to 3000m

c) Stream lines at 100m together with salinity field

Figure 5: *Additive projection of stream lines from an ocean simulation. Figure a) and b) show different depth layers, highlighting differences of the flow in both areas, c) shows a scalar field from the simulation (salinity of the water) blended with the stream line density (*$r_s = 1.0$*,* $e_f = 5$*).*

5. Discussion and Conclusion

Our method provides a simple way to reduce visual clutter in large trajectory sets and can be applied to all kinds of integration-based field lines (see e.g., [WHT12]). It emphasizes features, but does not highlight all types of flow features (e.g., saddle points). Vector fields that exhibit only small density variations or no asymptotic features (e.g., path lines in unsteady incompressible flows) require an adapted seeding strategy to emphasize features of interest. Merhof et al. [MSE*06a] mentioned visual artifacts for close-up views to line strip segments where the tangent direction is nearly parallel to the viewing direction. While they solved this problem by using additional point sprites, for density projections with a large amount of lines this seems to be not necessary. Future efforts can include the examination of the influence of special seeding structures toward the visible result, and the exploration of more advanced seeding and parameter selection strategies. Overall, our method provides a fast and interactive tool for the feature-oriented visual exploration of large vector field trajectory sets.

Acknowledgements We like to thank Simon Candelaresi (Astrophysics Group, Stockholm University) for providing the magnetic field data sets and Stephan J. Lorenz (Max-Planck-Institute for Meteorology, Hamburg) for providing the ocean simulation data set. This work was partially funded by the German Federal Ministry of Education and Research under grant number 01LK1213A.

References

[ATR*08] Annen T., Theisel H., Rössl C., Ziegler G., Seidel H.-P.: Vector field contours. In *Proc. Graphics Interface* (2008), pp. 97–105. 1

[CTJC10] Calamante F., Tournier J.-D., Jackson G. D., Connelly A.: Track-density imaging (tdi): Super-resolution white matter imaging using whole-brain track-density mapping. *NeuroImage 53*, 4 (2010), 1233 – 1243. 2

[EBRI09] Everts M. H., Bekker H., Roerdink J. B. T. M., Isenberg T.: Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1299–1306. 1

[EHS13] Eichelbaum S., Hlawitschka M., Scheuermann G.: LineAO – improved three-dimensional line rendering. *IEEE Transactions on Visualization and Computer Graphics 19*, 3 (March 2013), 433–445. 1

[GBWT11] Günther T., Bürger K., Westermann R., Theisel H.: A view-dependent and inter-frame coherent visualization of integral lines using screen contribution. *Proc. Vision, Modeling, and Visualization (VMV)* (2011), 215–222. 2

[LH11] Lampe O. D., Hauser H.: Interactive visualization of streaming data with kernel density estimation. In *Proceedings of the IEEE Pacific Visualization Symposium (PacificVis 2011)* (March 2011), pp. 171–178. 2

[LHD*04] Laramee R., Hauser H., Doleisch H., Vrolijk B., Post F., Weiskopf D.: The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum 23*, 2 (2004), 143–161. CGF. 1

[LMSC11] Lee T.-Y., Mishchenko O., Shen H.-W., Crawfis R.: View point evaluation and streamline filtering for flow visualization. In *Proc. IEEE Pacific Visualization* (March 2011), pp. 83–90. 2

[Lor05] Lorach T.: *Fake Volumetric Lines, NVIDIA SDK*. Tech. rep., NVIDIA Corporation, 2005. 2

[LT11] Lehmann D. J., Theisel H.: Features in continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization) 17*, 12 (2011), 1912–1921. 2

[MCHM10] Marchesin S., Chen C.-K., Ho C., Ma K.-L.: View-dependent streamlines for 3D vector fields. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (November 2010), 1578–1586. 2

[MJL*12] McLoughlin T., Jones M., Laramee R., Malki R., Masters I., Hansen C.: Similarity measures for enhancing interactive streamline seeding. *IEEE Transactions on Visualization and Computer Graphics* (Preprint 2012). 1

[MLP*10] McLoughlin T., Laramee R. S., Peikert R., Post F. H., Chen M.: Over Two Decades of Integration-Based, Geometric Flow Visualization. *Computer Graphics Forum 29*, 6 (2010), 1807–1829. 1

[MPSS05] Mallo O., Peikert R., Sigg C., Sadlo F.: Illuminated lines revisited. In *IEEE Visualization* (2005), pp. 19–26. 1

[MSE*06a] Merhof D., Sonntag M., Enders F., Nimsky C., Hastreiter P., Greiner G.: Hybrid visualization for white matter tracts using triangle strips and point sprites. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1181–1188. 1, 4

[MSE*06b] Merhof D., Sonntag M., Enders F., Nimsky C., Hastreiter P., Greiner G.: Streamline visualization of diffusion tensor data based on triangle strips. In *Bildverarbeitung für die Medizin 2006*, Handels H., Ehrhardt J., Horsch A., Meinzer H.-P., Tolxdorff T., (Eds.), Informatik aktuell. Springer Berlin Heidelberg, 2006, pp. 271–275. 2

[MSM*12] Machado G. M., Sadlo F., Müller T., Müller D., Ertl T.: Visualizing solar dynamics data. In *Proc. of the Vision, Modeling and Visualization Workshop (VMV)* (2012), Goesele M., Grosch T., Theisel H., Toennies K. D., Preim B., (Eds.), Eurographics Association, pp. 95–102. 2

[PF05] Pharr M., Fernando R.: *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation (GPU Gems)*. Addison-Wesley Professional, 2005. 2

[PPF*11] Pobitzer A., Peikert R., Fuchs R., Schindler B., Kuhn A., Theisel H., Matkovic K., Hauser H.: The state of the art in topology-based visualization of unsteady flow. *Computer Graphics Forum 30*, 6 (2011), 1789–1811. 1

[PYH*06] Park S. W., Yu H., Hotz I., Kreylos O., Linsen L., Hamann B.: Structure-accentuating dense flow visualization. *Eurographics /IEEE VGTC Symposium on Visualization* (2006), 163–170. 1, 2, 3

[SGS05] Stoll C., Gumhold S., Seidel H.-P.: Visualization with stylized line primitives. *IEEE Visualization Conference* (2005), 695–702. 1

[SLCZ09] Spencer B., Laramee R. S., Chen G., Zhang E.: Evenly spaced streamlines for surfaces: An image-based approach. *Computer Graphics Forum 28*, 6 (2009), 1618–1631. 1

[SWvdW*11] Scheepens R., Willems N., van de Wetering H., Andrienko G., Andrienko N., van Wijk J.: Composite density maps for multivariate trajectories. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (Dec. 2011), 2518 –2527. 2

[TMWS13] Tao J., Ma J., Wang C., Shene C.: A unified approach to streamline selection and viewpoint selection for 3D flow visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 3 (2013), 393–406. 2

[WCW*11] Wiebel A., Chan R., Wolf C., Robitzki A., Stevens A., Scheuermann G.: Topological flow structures in a mathematical model for rotation-mediated cell aggregation. In *Topological Methods in Data Analysis and Visualization*, Pascucci V., Tricoche X., Hagen H., Tierny J., (Eds.), Mathematics and Visualization. Springer Berlin Heidelberg, 2011, pp. 193–204. 2

[WHT12] Weinkauf T., Hege H.-C., Theisel H.: Advected tangent curves: A general scheme for characteristic curves of flow fields. *Computer Graphics Forum 31*, 2 (May 2012), 825–834. 3, 4

[WST*07] Weinkauf T., Sahner J., Theisel H., Hege H.-C., Seidel H.-P.: A unified feature extraction architecture. In *Active Flow Control*, King R., (Ed.), Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM). Springer, 2007, pp. 119–133. 3

[XLS10] Xu L., Lee T.-Y., Shen H.-W.: An information-theoretic framework for flow visualization. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (Nov.-Dec. 2010), 1216–1224. 1

[YWSC12] Yu H., Wang C., Shene C.-K., Chen J.: Hierarchical streamline bundles. *IEEE Transactions on Visualization and Computer Graphics 18*, 8 (August 2012), 1353–1367. 1

[ZSH96] Zöckler M., Stalling D., Hege H.-C.: Interactive visualization of 3D vector fields using illuminated stream lines. In *IEEE Visualization* (1996), pp. 107–113. 1, 3