Department: Dissertation Impact
Editor: Jim Foley, foley@cc.gatech.edu

This is the author’s preprint. The definite version is available here:
https://doi.org/10.1109/MCG.2019.2959568

Visibility, Topology and Inertia: New Methods in Flow Visualization

Tobias Günther
Department of Computer Science, ETH Zürich

***Abstract*—In this work, we address three different topics in scientific visualization. The first part introduces optimization strategies that determine the visibility of line and surface geometry, such that a balance between occlusion avoidance and preservation of context is found. The second part proposes new methods for the visualization of time-dependent fluid flows, including the accurate depiction of Lagrangian scalar fields, as well as a new category of vortex identification methods. The third part introduces finite-sized particles as new application area for flow visualization, covering geometry-based methods, particle separation, topology, vortex corelines and the determination of the origin of finite-sized particles.**

■ **FLOW VISUALIZATION** enables a deeper understanding of many dynamical systems, including the motion of the air that we breathe. Governed by the laws of physics, those systems shape our environment from the smallest scales, such as leaves circling in the air or smoke rising from a blown out candle, all the way to the largest global scales including hurricanes, tornadoes and tsunamis that affect billions of lives. Flow visualization thereby achieves something remarkable: it makes complex processes visible that are otherwise hidden from the naked eye. In the future, it will become more and more difficult to process flow data: Data sets are quickly growing in size, uncertainty is explicitly modeled, and many processes are time-dependent. Aside from the computational challenges, more thought is going into the careful selection of available information. The following sections study the selection of relevant lines and surfaces using a global optimization of opacity, considering valuable domain knowledge. Limiting the view to relevant aspects of the data not only supports visual exploration, it also significantly reduces the amount of data. Afterwards, we present two techniques for the feature extraction from unsteady flow. First, we introduce a technique that allows us to render Lagrangian scalar fields, such as finite-time Lyapunov exponents, at highest quality. Rendering images without grid discretization and step size errors during raymarching enables us to establish a ground truth that can serve as a baseline for other methods. Next, we propose a new class of vortex concepts that is invariant under uniform equal-speed rotations of the reference frame. Finally, we extend traditional flow visualization methods, which typically only consider the observation of massless particles, to the study of finite-sized, so-called inertial particles. We extend common concepts, such as integral curves, finite-time separation, vortex corelines, critical points, and asymptotic flow behavior to the inertial case, and present a novel approach to the source inversion problem. The study of inertial particles opens a vast field of visualization applications, such as sediment transport, pollution problems or view restriction during landing maneuvers of aircraft. We begin with the visibility optimization problem.

VISIBILITY OPTIMIZATION

When visualizing 3D fluid flows, we often need to deal with occlusion problems, when dense sets of lines or surfaces are rendered. If geometry is removed without considering its significance in the image, meaningful information will be lost - either because it is missed or because it is simply hidden behind less relevant geometry. With growing data sizes and simulation setups, this selection problem will become more severe, prompting more careful consideration of which data to show in order to support fluent and informative interactions and exploration.

Opacity Optimization of 3D Line Fields

The careful consideration of which geometry to show is often a user-centered process. Whenever a domain scientist views a new data set, he or she might have a particular goal in mind, e.g., the assessment of aneurysm rupture risk in medical blood flow data, the identification of vortices in ocean flows or the observation of flow near topological features of interest. In our first work [1], we give users an instrument to characterize their interest in a given piece of line geometry. To this end, we introduce a scalar attribute that is defined per vertex, which we call *importance*. Our goal is to find an opacity value for each vertex, such that relevant information is not occluded. To reduce the problem size, we discretize all lines into segments and first compute an optimal opacity per segment, which is then interpolated onto the vertices. Deciding on a useful opacity assignment requires a measure of quality. In the context of optimization problems, this is called an energy. Our energy considers a number of terms, each being responsible for a certain desirable property. The first term is a regularization that makes all lines as visible as possible. The next terms require a notion of occlusion. This means, we need to quantify how much one line segment occludes another one, which we estimate by rasterizing all lines, sorting their fragments into pixel linked lists and sorting the lists. Using this relatively simple data structure, we iterate the fragments that are rasterized into a pixel and count how often a fragment of one line segment occludes another fragment of another line segment. Our second energy term then penalizes unimportant line segments in front of important line segment, forcing the opacity of the segment in front to be reduced. The weight of this energy term effectively steers the overall amount of opacity in the image. Further, we found it useful to remove unimportant line segments behind important ones, which is achieved by the third energy term. The removal of background clutter increases the visual contrast of the relevant information, making it more readily seen. Finally, the last energy term smoothes the resulting opacities along the lines, making sure that opacity varies smoothly. The resulting energy function is quadratic and can be solved with a constraint linear optimization, since the opacity values range between zero and one. The optimization is solved asynchronously on the CPU and the opacity solution is streamed to the GPU for display. Examples of our interactive visibility optimization can be seen in **Figure 1**.

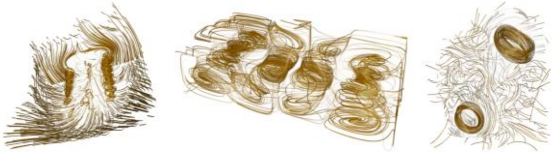

Figure 1. Examples of our interactive global line visibility optimization, revealing the user-defined interesting structures in the data sets: vortices from the wake of a helicopter (left), four convection cells (middle) and field lines of a magnetic field (right). (Used, with permission, from [1].)

Opacity Optimization of Time-dependent Lines

The visibility optimization of time-dependent line geometry places harder constraints to maintain interactivity. First, the runtime of the optimization must be improved, since the line geometry is changing quickly, invalidating the results that have been computed a few moments ago. Second, the optimization must be temporally coherent, which means that the transparency should vary smoothly over time. The runtime of the visibility optimization largely depends on the number of line segments, which corresponds to the number of unknowns in the constraint linear optimization problem. Looking at a typical 3D scene, we find that some lines are not visible on the viewport and that some geometry is far away from the camera, requiring less detail. To automatically adjust our line discretization, i.e., to distribute our computational budget better, we precompute a hierarchical line discretization that can be traversed at runtime [2]. Based on the camera position and orientation, we determine an optimal cut in the line hierarchies to find a suitable discretization resulting in roughly equally-sized line segments in image space. Further, the occlusion estimation can be accelerated by constructing the pixel linked lists at lower screen resolution. In order to obtain temporal coherence, we introduce a temporal smoothing term, which requires us to establish a correspondence between the lines of the previous and the current time steps in order to match the opacity of the previous frame closely. The correspondence can be established in two ways: either by advection with the flow of by pinning the vertex indices. None of these two heuristics is perfect, as they both work best in different situations. The optimal choice is therefore application-dependent. Temporal changes in the hierarchy cut can be resolved by propagating the previous opacity solutions up and down in the line hierarchies before establishing the correspondence. With the help of the accelerations, we are able to optimize the visibility of time-dependent lines in a frame-coherent way. Examples are shown in **Figure 2**.

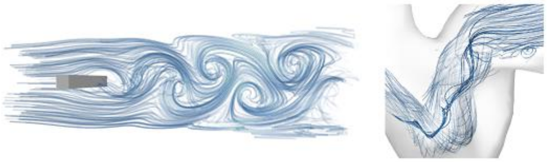

Figure 2. Visibility optimization of time-dependent line geometry: streaklines in a von-Kármán vortex street (left) and the vortex coreline of blood flow in a vessel (right). (Used, with permission, from [2].)

Opacity Optimization of Surfaces

Next, we move on to visibility optimization of surface geometry. The procedure is very similar to the previous steady line geometry method. We subdivide all surfaces into patches, setup an energy that measures the quality of an opacity assignment and then solve the linear constraint optimization problem. While the subdivision of lines into equally-sized segments was rather straightforward to do in linear time complexity, the subdivision of surfaces is computationally more expensive. First, we distribute a number of points on the surfaces using a geodesic farthest point sampling. The geodesic distances are computed with the geodesics in heat method. The result is a point set, which serves as input to a Voronoi diagram computation. Each surface element of the Voronoi diagram has approximately equal surface area and can serve as patch in the optimization. A disadvantage of transparency-based visualizations is an impairment of spatial perception, which is why we include a contrast enhancement at the silhouettes. The silhouettes are likewise part of the visibility optimization and help in the perception of layer order. Examples of our method are shown in **Figure 3**.

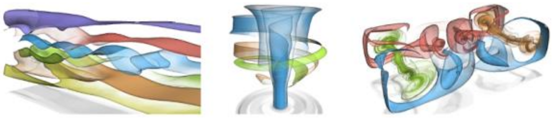

Figure 3. Visibility optimization of surface-based flow visualizations, showing a vortex street (left), a static mixer (middle) and convection cells (right). (Used, with permission, from [3].)

Impact of Visibility Optimization

The line of research on visibility optimization naturally extends toward point and volume data. With the subsequent volume visualization work of Ament et al. [4], the energy was reformulated into a pixel-based energy that can be efficiently minimized on the GPU. More recently, the scalability of the geometry-based algorithms was improved by substituting the fragment linked lists with Fourier approximations [5] in the pixel-based formulation, enabling the interactive exploration of large geometric data sets within fixed memory bounds. Visibility optimization became part of a larger trend toward automated visualization. Aside from optimal transparency, the automatic choice of colors, viewpoints and other visualization parameters became a recent trend, bridging the gap between sophisticated visualization algorithms and the usability constraints in the actual applications.

FEATURES IN TIME-DEPENDENT FLOWS

The development of flow analysis tools for time-dependent vector fields was and still is a hot topic in scientific visualization, due to the many unsolved research problems. In the following sections, we utilize light transport simulations from offline rendering to render scalar fields that were computed from particle advection at an unprecedented accuracy. Then we introduce a new class of vortex definitions that are applicable in situations when vortices move along circular paths around a known center of rotation, such as in mixing devices, hydrocyclones or centrifugal pumps.

Monte Carlo Rendering of Separations

In time-dependent flows, the temporal domain is usually limited by the available data, which is why we cannot assess asymptotic behavior as in steady flows. This creates a serious problem, since none of the topology-based visualization methods of steady flows is applicable in time-dependent flows. Instead, we study so-called Lagrangian coherent structures (LCS), which only consider a certain finite time window, including vortex boundaries (elliptic LCS), jet cores (parabolic LCS) and separating structures (hyperbolic LCS). A common approximation to the hyperbolic LCS is the finite-time Lyapunov exponent (FTLE), which is a simple measure of the expansion of a small sphere during particle advection. In forward time, the separation is largest on repelling structures, whereas in backward time, the largest separation reveals attracting structures. Since the FTLE field arises from particle integration, the ridges (lines in 2D and surfaces in 3D), can become arbitrarily thin, which creates a sampling problem: both the discretization onto a regular grid, as well as the finite steps during direct volume rendering will create artifacts. In our work [6], we utilize Monte Carlo methods from physics and offline production rendering to render the FTLE fields in an unbiased manner. For this, we treat the volume rendering of the FTLE field as a light transport problem in an inhomogeneous participating medium with single-scattering. This means that each photon that exists the light source is expected to bounce only once in the medium, at the so-called scattering point, before reaching the viewer's eye. In this formulation, we need to estimate how much light is absorbed on the way from the light source to the scattering point and then from the scattering point to the viewer. Both transmittance values are estimated stochastically using an unbiased sampling technique called Delta tracking. The iterative rendering algorithm is free of discretization errors and produces visualizations, such as the one shown in **Figure 4**. At present, the avoidance of artifacts is bought at the expense of a very high computation time of up to days for a single high-resolution image. The visualizations can serve as ground truth for faster approximate techniques and have been used by us in multiple occasions for science communication.

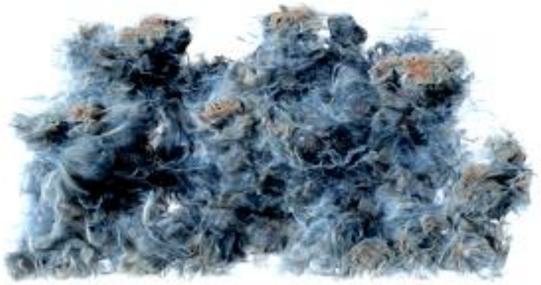

Figure 4. Monte Carlo rendering of the separating structures in a convection simulation of cumulus clouds. (Used, with permission, from [6].)

Rotation-invariant Vortex Measures

Next, we investigate features in rotating devices, such as centrifugal pumps or rotating mixers. In such systems, vortices are studied to assess mixing properties and the potential trapping of particles. A general problem of feature extraction from time-dependent vector fields is the dependence of the observed features on the movement of the observer. Imagine a cylindrical container in which a rotor propels a fluid into a rotating motion. An observer that is standing next to the flow will see very different structures than an observer that is holding on to the rotor and is spinning with it. In our work [7], we propose vortex extraction methods that are invariant to the choice of the rotating reference frame. To this end, we utilize methods that are invariant to translating reference frames. By transforming the domain from a Cartesian grid into polar coordinates, the rotating motion in the Cartesian space becomes a translation in polar coordinates along the angle dimension. This way, Galilean invariance, i.e., the invariance to equal-speed translations of the observer, directly translates to rotation invariance, i.e., the invariance to equal-speed rotations of the observer. We derived expressions for the calculation of the velocity and Jacobian matrix in the rotating frame or reference for a given center of rotation and applied the method to the extraction of region-based and line-based vortex measures. **Figure 5** shows examples of a turbulent flow in a rotating mixer and vortex corelines, i.e., the lines around which other particles are rotating in a centrifugal pump. A current limitation is that the center of rotation must be known in advance.

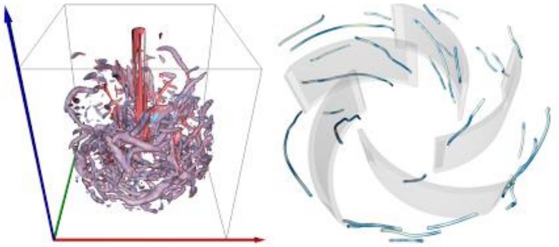

Figure 5. Visualization of rotation-invariant vortex measures, for region-based criteria in a rotating mixer (left) and line-based criteria in a horizontal section of a centrifugal pump (right). (Used, with permission, from [7].)

Impact of Feature Extraction

In recent years, we applied the Monte Carlo rendering method of the finite-time Lyapunov fields to different data sets, including meteorological data, asteroid impact simulations, and dark matter simulations from cosmology. More recently, we improved the performance by a factor of about eight by substituting the binary transmittance estimator with ratio tracking, by using gradient domain rendering and by introducing Fourier approximations of the light transmittance. With these improvements, we were able to render videos of the finite-time Lyapunov exponent, including temporal animations, slicing and camera navigations along pre-determined paths.

More recently, we and also other research groups developed different approaches for feature extraction and unsteady topology. First, we extended the rotation invariance to a more general method that locally optimizes for a rotation and translation of the observer in which the flow becomes steady, without having to know the center of the rotation [8]. The method thereby becomes objective, which is a desirable formal property that guarantees that vortex measures give the same result if the vortices move along smooth rotating or translating paths. Similarly, Hadwiger et al. [9] formulated a global optimization problem that also allows users to align vortices in space-time. Aside from traditional optimization-based methods, we trained a deep neural network [10] to recover the optimal reference frame even in the presence of noise or resampling artifacts, which greatly improves the numerical robustness. Up until this point, the reference frame optimizations were only used for vortex extraction. Baeza Rojo et al. [11] extended the work in two ways: First, they generalized the class of transformations to spatially-varying displacement transformations, which include all previous local methods as special case. Second, the optimization decomposes the flow into two parts: a part that contains the flow features and a part that describes the ambient motion of the features. They proposed that the first part is not only useful for vortex extraction, but also for other topological elements and features, such as hyperbolic trajectories. In principle, this opens the question whether or not it is possible to extract time-dependent topology locally per time slice or whether features are derived from the observations of particles over a longer time range, which also received more attention.

FINITE-SIZED PARTICLES IN FLOWS

Finally, we introduce finite-sized particles as another application area for flow visualization. In the past, most research in flow visualization concentrated on the motion of massless particles, which are also known as tracer particles, since they follow the flow tangentially. This makes them perfect tools for the study of the vector field itself. In many applications, however, the mass and the resulting inertia of particles play a significant role, for instance when studying the uplift of dust and sand when helicopters or airplanes approach a dusty landing zone. The description of motion of these finite-sized particles uses second-order ordinary differential equations, which requires the consideration of both the particle position and the particle velocity to evaluate where the particle goes next. Therefore, both the state of a particle and the vector field that describes its motion are higher-dimensional, which creates interesting computational and visualization challenges.

Mass-Dependent Separation

The behavior of inertial particles is determined by their size and density. The motion therefore becomes parameter-dependent, prompting questions about the critical parameter thresholds that lead to differences in the behavior. For instance, for which mass threshold does an inertial particle in the airflow around a helicopter get dragged back into the rotor disk? After studying the geometry of particle trajectories, we set out to visualize how the separation of inertial particles depends on their size and density. Inspired from the finite-time Lyapunov exponent, which measures separation due to a change in the initial position, we introduce the finite-time mass separation (FTMS), which similarly measures the separation due to a change in a particle parameter, such as the particle size [12]. To visualize the behavior, we created three coordinated views that show different aspects of the particle motion. All three views are shown in **Figure 6**. In all images, the color corresponds to the particle size. The left view displays the particle size for which the strongest separation occurred. The lines of strongest separation do not coincide with the repelling structures of the underlying flow, which is another aspect we followed up on later. The central view visualizes the continuous spectrum of trajectories in the space-time domain for a range of different particle sizes. We can see that a separation point exists that decides whether a particle moves left or right, as the particle approaches the wall. This gave us a useful hint that the topological structures in inertial flows are size-dependent. The right view shows how the particles separate from each other. This view allows the user to quickly identify the critical particle sizes around which the behavior of particles changes. Pixels for seeding and individual trajectories can be clicked and are highlighted in all views accordingly.

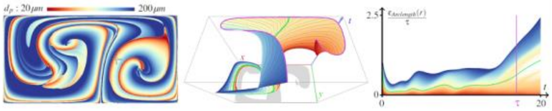

Figure 6. Three linked views, showing different aspects of size-dependent particle separation: the maximum separation in the spatial domain (a), the separation manifold in space-time (b), the evolution of frontline distances (c). (Used, with permission, from [12].)

Inertial Steady Topology in 2D Flows

The previous section showed us that the motion of inertial particles is strongly dependent on the particle size. Naturally, the asymptotic motion also differs, which means that the topological elements in steady vector fields will be different. In our work [13], we first identify the locations of critical points, i.e., the locations at which the high-dimensional flow that governs inertial particle motion becomes zero. The critical points are then classified by the flow behavior near them, which is achieved by inspecting the eigenvalues of the Jacobian of the high-dimensional flow. Due to the general structure of the Jacobian, it turns out that sources cannot exist in the high-dimensional flow, since at least two eigenvalues will always be negative. This is an indicator that an attracting force is omnipresent, which will play a role later. With only sinks and saddles existing, we can only compute stable sets, i.e., subspaces of the high-dimensional state space that lead particles into the same sink. Which sink is entered depends on both the seeding position and the seeding velocity. To visualize both variables at the same time, we use a multi-dimensional stacking, as illustrated in **Figure 7** for three data sets. First, the spatial domain is sampled with a set of points, which are placed on a hexagonal grid. Each grid point represents a spatial seed point. Around each of the points, we span a small disc. The pixel location in the disc encodes the initial velocity of the particle. For instance, the pixel at the center of a disc corresponds to a particle released from rest, whereas the right-most pixel represents a right-bound initial velocity. Since the velocity domain is represented by discs, we chose the hexagonal grid layout, since this is the densest circle packing. With this, pairs of initial position and initial velocity are represented in the scene and we can trace inertial particles to record their asymptotic behavior. In the images, the sink that is reached is color-coded, effectively coloring the stable sets in the same color. To facilitate the user interaction, the circle glyphs are computed progressively over multiple frames. The selected glyph is enlarged in the bottom right corner and users can seed inertial particle trajectories (black curves) by clicking into the enlarged glyph. In the images, we also place icons at the locations of sinks (blue) and saddles (yellow and orange).

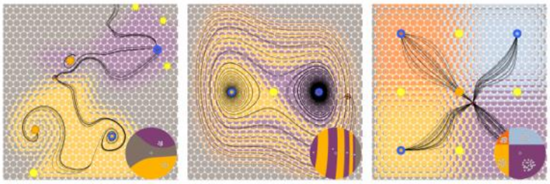

Figure 7. Critical points and color-coding of stable sets for selected initial positions (circle position) and initial velocities (position in circle). Here for three synthetic flows: a magnetic field (left), an oscillator (middle) and a conservative analytic flow (right). (Used, with permission, from [13].)

Source Inversion of Inertial Particles

In the previous section, we observed that the Jacobian of the high-dimensional flow always contains at least two negative eigenvalues. The fluid dynamics literature tells us that this is the result of an omnipresent global attracting manifold that all inertial particles are asymptotically drawn too. Because of this attraction in forward direction, we can expect a repelling force in backward direction. This force is very strong, making the backward integration of an inertial particle very difficult. This is the so-called source inversion problem. Even when tracing a trajectory forward for a while and then trying to trace it backward for the same time, the tiniest numerical integration error will amplify, quickly shooting the backward trajectory away at exponential rates. In practice, this means that it is very difficult to estimate the origin of pollutions in air or water. In our work [14], we reformulate the problem to propose a new approach. Instead of specifying the initial position and initial velocity of the backward integration, we set the initial position and the velocity reached after the backward integration. Figuratively speaking, we could ask for all locations that a pollutant came from, when it was dropped with a certain velocity into the air or water. Given a certain location, we can inspect the behavior of nearby particles to see in which direction we would need to move next such that a slightly longer inertial particle integration will take us to the same destination. The integration is computationally expensive, and numerically still quite challenging for longer integration durations or when inertial particles oscillate, but it generally helped us to trace inertial particles farther back in time. The set of possible seed points leading to the same destination assembles curves, which are shown in blue in **Figure 8**.

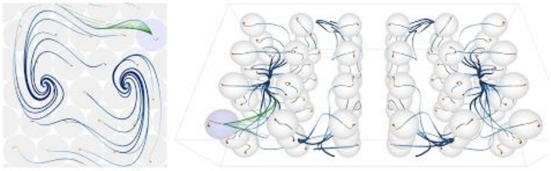

Figure 8. Visualization of sources (blue) of finite-sized particles at certain observation points (gray circles/spheres). The sources assemble lines that can be traced as integral curves, here shown for 2D (left) and 3D (right). (Used, with permission, from [14].)

Vortex Corelines of Inertial Particles

Finally, we concentrated on the extraction of vortex corelines of inertial particles, which are the lines around which other particles are rotating. In traditional massless flows, those lines are found as the solution to the so-called parallel vectors operator, which requires two vector fields to be parallel. A standard criterion for inertial particles is to request that the particle velocity be parallel to the acceleration. Both properties are local and thus the lines can be solved for in parallel for all voxels of the domain. Similar to the critical points above, we can apply the same vortex coreline criterion to the high-dimensional flow in which the inertial particles live. This would result in a parallel vectors search in a six-dimensional domain if the flow is steady and to a seven-dimensional domain if it is unsteady. On first sight, this seems to be computationally very expensive and almost infeasible for large domains. However, we showed [15] that the criterion can be relaxed by requesting parallelism only in the position and velocity subspaces, which simplifies the problem in both the steady and the unsteady case to a three-dimensional parallel vectors problem. The computation therefore becomes as expensive as the search for vortex corelines of massless particles. Examples of inertial corelines are shown in **Figure 9**.

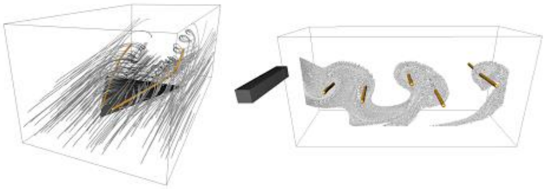

Figure 9. Vortex corelines of finite-sized objects in the flow around a delta wing (left) and in the von-Kármán vortex street in the wake of an obstacle (right). (Used, with permission, from [15].)

Impact of Inertial Particles

We could see already at the example of our finite-time mass separation that the separating structures are mass-dependent. To find attracting structures of inertial particles, we applied our source inversion approach, showing much better correspondence to preferential particle settling than previous massless approaches. To better understand the mass-dependent differences in particle behavior regarding the attracting manifold, Baeza Rojo et al. [16] visualized the asymptotic particle motion of unsteady 2D flows in lifted 3D domains. The paths of feature extraction and inertial particles have further merged, when we proposed objective vortex criteria of inertial particle motion [17]. Inertial particles are only one example of motion in a higher-dimensional vector field. In the meantime, more work on the visualization of general high-dimensional vector fields appeared, including topology and camera projections. Further extensions and applications to other dynamical systems will provide interesting testing grounds for the study of high-dimensional flows. While some concepts are generalizable to all higher-order flows, we noticed that some classes of dynamical systems have very particular properties that can be analyzed and utilized to build more meaningful and interactive visualization.

CONCLUSION

This thesis contributed a number of geometry-based, feature-based and topology-based flow visualization methods for traditional massless and higher-dimensional inertial flows. The first part of the thesis introduced optimization methods that adjust the visibility of line and surface geometry such that relevant information is seen for a given viewpoint. The second part introduced feature extractors for time-dependent flow, including a Monte Carlo rendering method of finite-time Lyapunov fields and rotation-invariant vortex criteria. The last part discussed challenges and opportunities of finite-sized particle motion, covering separation, topology, source inversion and vortex coreline extraction.

There are still many challenges to address. To improve the interactive navigation through data, especially in larger domains, we need more automation and user guidance. The flow feature extraction from time-dependent flows is still an actively research problem, with many topics including uncertainty, vortex cascades, scalability, and unsteady topological elements in general. In the realm of high-dimensional flows, there are more applications to explore and methods to extend from the massless case.

ACKNOWLEDGMENT

The author would like to express his deepest gratitude to his PhD advisor Holger Theisel, who supported me before, during and after his thesis work.

■ REFERENCES

1. T. Günther, C. Rössl, and H. Theisel, “Opacity Optimization for 3D Line Fields”, ACM Transaction on Graphics (Proc. SIGGRAPH), vol. 32, no. 4, pp. 120:1–120:8, 2013.

2 T. Günther, C. Rössl, and H. Theisel, “Hierarchical Opacity Optimization for Sets of 3D Line Fields”, Computer Graphics Forum (Proc. Eurographics), vol. 33, no. 2, pp. 507–516, 2014.

3. T. Günther, M. Schulze, J. Martinez Esturo, C. Rössl, and H. Theisel, “Opacity Optimization for Surfaces”, Computer Graphics Forum (Proc. EuroVis), vol. 33, no. 3, pp. 11–20, 2014.

4 M. Ament, T. Zirr, and C. Dachsbacher, “Extinction-Optimized Volume Illumination”, IEEE Transactions on Visualization and Computer Graphics, vol. 23, no. 7, pp. 1767–1781, 2017.

5. I. Baeza Rojo, M. Gross, and T. Günther, “Fourier Opacity Optimization for Scalable Exploration”, IEEE Transactions on Visualization and Computer Graphics, in print, 2020.

6. T. Günther, A. Kuhn, and H. Theisel, “MCFTLE: Monte Carlo Rendering of Finite-Time Lyapunov Exponent Fields”, Computer Graphics Forum (Proc. EuroVis), vol. 35, no. 3, pp. 381–390, 2016.

7. T. Günther, M. Schulze, and H. Theisel, “Rotation Invariant Vortices for Flow Visualization”, IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE SciVis), vol. 22, no. 1, pp. 817–826, 2016.

8. T. Günther, M. Gross, and H. Theisel, “Generic Objective Vortices for Flow Visualization”, ACM Transaction on Graphics (Proc. SIGGRAPH), vol. 36, no. 4, pp. 141:1–141:11, 2013.

9. M. Hadwiger, M. Mlejnek, T. Theussel, and P. Rautek, “Time-Dependent Flow seen through Approximate Observer Killing Fields”, IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE SciVis), vol. 25, no. 1, pp. 1257–1266, 2019.

10. B. Kim and T. Günther, “Robust Reference Frame Extraction from Unsteady 2D Vector Fields with Convolutional Neural Networks”, Computer Graphics Forum (Proc. EuroVis), vol. 38, no. 3, pp. 285–295, 2019.

11. I. Baeza Rojo and T. Günther, “Vector Field Topology of Time-Dependent Flows in a Steady Reference Frame”, IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE SciVis), in print, 2020.

12. T. Günther and H. Theisel, “Finite-Time Mass Separation for Comparative Visualizations of Inertial Particles”, Computer Graphics Forum (Proc. EuroVis), vol. 34, no. 3, pp. 471–480, 2015.

13. T. Günther and H. Theisel, “Inertial Steady 2D Vector Field Topology”, Computer Graphics Forum (Proc. Eurographics), vol. 35, no. 2, pp. 455–466, 2016.

14. T. Günther and H. Theisel, “Source Inversion by Forward Integration in Inertial Flows”, Computer Graphics Forum (Proc. EuroVis), vol. 35, no. 3, pp. 371–380, 2016.

15. T. Günther and H. Theisel, “Vortex Cores of Inertial Particles”, IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE SciVis), vol. 20, no. 12, pp. 2535–2544, 2014.

16. I. Baeza Rojo, M. Gross, and T. Günther, “Visualizing the Phase Space of Heterogeneous Inertial Particles in 2D Flows”, Computer Graphics Forum (Proc. EuroVis), vol. 37, no. 3, pp. 289–300, 2018.

17. T. Günther and H. Theisel, “Objective Vortex Corelines of Finite-sized Objects in Fluid Flows”, IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE SciVis), vol. 25, no. 1, pp. 956–966, 2019.

Tobias Günther joined the Computer Graphics Laboratory (CGL) at the ETH Zürich as a postdoctoral researcher in 2016. He received his M.Sc. in Computer Science in 2013 and his Dr.-Ing. (Ph.D.) in 2016 both from the Otto-von-Guericke University of Magdeburg. His research interests include scientific visualization, progressive light transport, real-time rendering. Contact him at tobias.guenther@inf.ethz.ch.