Conclusion

The visualization of vector fields is an established, however still ongoing field of research. The goal of this thesis was to develop techniques that allow the visualization and analysis of various kinds of vector fields that encode directional information, being it flow or another property. Depending on the nature of particular fields, different technical or fundamental problems had to be solved to achieve this goal.

For highly complex and memory-consuming vector fields, a cluster environment was employed to create an interactive visualization tool as detailed in Chapter 2. With each compute node that is integrated into the cluster environment, the available memory increases directly, however, the expected speed-up does only scale well for a small number of nodes, but with decreasing effectiveness for a higher number of cluster nodes. More precisely, the performance gain decreases with every added cluster node due to additional communication overhead. Further research is necessary to reduce the communication overhead to allow many cluster nodes being involved in the visualization process. Additionally, the parallel rendering method could be extended to the projection of the surface geometry instead of replicating the mesh on each node. This would allow visualizing extremely large surface meshes.

For the analysis of vector fields, a visualization tool not only needs to be interactive, it also must be able to guide the user and the visualization metaphors that are used must be helpful and efficient. For a vector field, not only the direction of the flow is of interest, but also the magnitude which can be visualized with various visual cues, with animation being among them. In order to employ results of cognitive psychology research, Chapter 3 presents a technique that was developed to visualize vector fields in a dense way using animation. In contrast to existing methods, this technique is able to tune its pattern frequencies to achieve the optimum for the human visual system. Recently, Yeh et al. published a technique to visualize stream lines using repeated asymmetric patterns [YLL12] based on the same idea of using patterns that are orthogonal to the stream lines that are visualized. Although these methods do not lend themselves directly to an extension to true 3D fields, Schulze et al. [SRGT12] present a solution based on “as-perpendicular-as-possible sur-
faces”, which can be seen as the counterpart of ortho-vis in 3D. What remains to be done is an extensive user study to confirm the usefulness and effectiveness of the proposed method for 2D and 2.5D fields. Additionally, the focus of ortho-vis lies on low-level, local motion perception. Therefore, the relationship between global motion perception and the effectiveness of conveying flow structures remains an open question. Further investigations are also needed to make sure that the visual signatures introduced by the temporal filtering process give the correct impression of the visualized vector field.

The second part of this thesis takes a different approach on the visualization of vector fields, as features of interest are used to simplify the resulting images—visual clutter and complexity is reduced by using a topological approach to visualization. Such “simplification” methods are gaining importance more and more, since data sets are challenging not only in terms of memory size, but also with respect to complexity which manifests itself, e.g., in the form of turbulence—a phenomenon which is extraordinarily hard to visualize in an easily understandable way, and especially hard to find are cues that help the user analyze the inner workings of turbulence. The technique described in Chapter 4 combines existing techniques—LIC on curved surfaces (described in Chapter 1.4.5), and previous work by Sadlo and Weiskopf [SW10]—to enhance the visualization of LCS. This combination of techniques gives simple representations for data sets that are not very turbulent. Perception problems can, however, arise for complicated flow fields with turbulent regions. The structure of turbulent flows is highly complex by its nature, therefore, future work could address techniques that reduce the complexity of the visualization by finding ways to visualize only the essence of such complex data. Other future work is the extension to 3D time-dependent vector fields, i.e., space-time visualization in four dimensions which includes the intersection curves of LCS and the surfaces they span over time.

Guiding the user with visualization metaphors that are designed to simplify the analysis of vector fields was also the goal for the visualization of magnetic flux in magnetostatic fields presented in Chapter 5. Available techniques from classical vector field topology had to be extended since they were not directly applicable for this specialized scenario. Here, a topological construct, the connectrix, was introduced that is designed to visualize regions that are connected with each other with respect to magnetic flux—as opposed to classical topology which uses separatrices to visualize where regions of different flow behavior are located. Relevant results for application domain collaborators were obtained with this method. An open question left for future work is the extension to three dimensions. The main challenge here will be that the scalar-valued potential $A$ will have to be extended to a 3-component vector field potential.

The third and final part of this thesis introduces continuous scatterplots, a statistical visualization method that was created to analyze scientific data sets. As
opposed to traditional scatterplots, this method allows one to work with data defined on a continuous domain based on a generic mathematical model. This mathematical model maps an arbitrary density value defined on an $n$-D input data set to $m$-D scatterplots. Not only does this model provide a solid and reliable basis for many variants of frequency plots of continuous data, but it also allows one to assess the errors introduced by previous discrete frequency plots, which can be viewed as examples of numerical approximation of continuous scatterplots. Therefore, continuous scatterplots lead to the same basic visual mapping as traditional histograms, scatterplots, or other frequency plots. In this way, they add one missing piece to the general approach of applying statistical and information visualization methods to scientific data. Furthermore, the generic model presented in this chapter has value of its own in any scientific discipline that strives for unification and simplification.

Several ways for the implementation of continuous scatterplots are explored that either broaden the possibilities in terms of interpolation or reconstruction methods that can be used to compute a continuous scatterplot, or decrease the computational cost to create such a plot. To achieve the latter, speeding up the computation of continuous scatterplots is attained with several different computation schemes that employ user-controlled approximation methods to reduce the time to compute a continuous scatterplot. Finally, hardware acceleration is used to reach the same goal utilizing the parallel architecture of GPUs. Remaining future work is the extension to higher-dimensional spatial domains, such as time-dependent 3D data sets.

The mathematical model of continuous scatterplots provided new possibilities for follow-up research in this direction, leading to a several related papers, e.g., “Continuous Parallel Coordinates” by Heinrich and Weiskopf [HW09], and “Discontinuities in Continuous Scatterplots” by Lehmann and Theisel [LT10]. These examples originate from the visualization community, however, researchers of other communities work on related topics. To name an example from computer vision, Dowson et al. [DKB08] construct a continuous model to obtain the joint distribution of image pairs. Related to this work is the paper by Kadir and Brady [KB05] that addresses the problem of estimating statistics in regions of interest by applying continuous density estimates.

The implementations of continuous scatterplots presented in the third part of this thesis are not the only possible approaches—due to the generic mathematical basis, the technique presented in Chapter 6 is not only unrestricted with respect to the dimensionality of the data that it handles, it is also open to various implementation approaches. Although several methods have been developed to compute continuous scatterplots, e.g., as presented in Chapters 7 and 8, improvements with respect to computational performance or integration quality are still possible. Therefore, it becomes clear that the true value of this approach is not technology-based or hardware-based—something that may be outclassed sooner or later by future technology—the main contribu-
tions are found in the theoretical foundation presented in this thesis.

During the course of my thesis, I have encountered problems that are related mostly, but not only to flow visualization. It became clear that even software that is written to create a proof-of-concept application needs a profound software engineering approach to avoid problems that are related to organically grown software. However, extensive application of software engineering principles prolong the development process of proof-of-concept software too much—a good balance between the extremes is necessary to produce software of as-high-as-possible quality without losing too much time. For this reason, the visualization techniques described in Chapter 6 and 8 were implemented using the MegaMol framework developed by Sebastian Grottel. This framework encourages and supports high code reusability as well as a modularized approach to software engineering.

For the remaining visualization techniques of this thesis, individual tools were developed using the most up-to-date software and hardware technologies that were deemed best suitable to solve the technical problems related to the respective research project at this time. Although these tools are technically different, they allow the interactive analysis of vector fields from different points of view—perception-oriented with the methods presented in Part I, feature-based analysis with techniques of Part II, and, lastly, focusing on multi-attribute, data-based analysis in Part III.

Despite their technical differences, these proof-of-concept tools can be seen as modules that could be integrated in a larger visualization system. Depending on user requirements, this would allow, e.g., to visualize large, complex vector fields using the cluster environment described in Part I, and combine it with topological methods presented in Part II to handle occlusion and ease the analysis of such a vector field.

Closing the link to Section 1.1, the contributions to the challenges mentioned there are critically evaluated. Every method presented in this thesis adds some details to the overall picture. Some techniques are more generally applicable, whereas others are more application specific. A critical sum-up based on the remarks of this section is given in Figure 9.1, which is adopted from Section 1.8 and modified to reflect the additional aspect of generality and application specificity.

On a final note, and as mentioned in the beginning of this chapter, the visualization and analysis of vector fields is a still ongoing field of research. However, the visualization community loses interest in techniques that concentrate merely on directly visualizing vector fields, as it is the case for the ones presented in Part I. What are the reasons for this development? Possibly, this is due to the high saturation of well working visualization methods. These methods have arrived at a very high level, leaving not much room for improvements. Because of this, and because of the need for methods that are
Figure 9.1: Visualization of the techniques presented in this thesis with respect to level of abstraction and, as a second aspect, their generality or application specificity.
able to handle even more complex data sets, the attention turns to techniques that emphasize features, like, e.g., the ones presented in Part II of this thesis. The importance of models for such techniques, as well as corresponding algorithms to compute visual representations, is expected to rise even more in the future.
Bibliography


[BSW+12] S. Bachthaler, F. Sadlo, R. Weeber, S. Kantorovich, C. Holm, and


1991. 107


166 Bibliography

64.1–64.10, 2005. 157

Physical Chemistry Chemical Physics, 10:1883–1895, 2008. 96, 108


[SEP+01] G. Stoll, M. Eldridge, D. Patterson, A. Webb, S. Berman, R. Levy,


logical Methods in Data Analysis and Visualization, pages 151–165. Springer, 2011. 79


[Wei06] D. Weiskopf. GPU-Based Interactive Visualization Techniques. Springer, 2006. 18


Curriculum Vitae

Sven Bachthaler


Sven Bachthaler graduated in 2005 from the Universität Stuttgart, Germany, with the Dipl. Inf. (MSc) degree. In 2006, he started with the graduate program of computing science at Simon Fraser University, Canada. In 2007, he returned to Germany to continue working on his PhD degree at the Visualization Research Center, Universität Stuttgart (VISUS). His research interests include scientific visualization and computer graphics.