# FAnToM - Lessons Learned from Design, Implementation, Administration, and Use of a Visualization System for Over 10 Years

Alexander Wiebel*
Max Planck Institute for Human Cognitive and Brain Sciences

Christoph Garth†
University of California, Davis

Mario Hlawitschka‡
University of California, Davis

Thomas Wischgoll§
Wright State University

Gerik Scheuermann¶
Universität Leipzig

## 1 Introduction

Scientific visualization has become a central tool in many research areas since it has been established as a research discipline in 1987 [2]. Naturally, this development resulted in software tools specifically tailored for the visualization task at hand. While many such tools exist, the design choices underlying them vary greatly.

This abstract describes some aspects of the FAnToM[1] visualization system that is being developed since 1999. Initially created to support research in topological methods for vector and tensor fields, the system quickly grew into a visualization platform for general flow visualization specialized to data represented on unstructured grids. From this origin, FAnToM derives advanced data structures for point location and interpolation over unstructured meshes, as well as fast integral curve capabilities. More recently, FAnToM has gradually been extended to serve a wider area of visualization applications, including medical and graph visualization. Throughout the development of FAnToM, close collaboration with application domain scientists has been a strong priority to facilitate the system's usefulness on state-of-the-art problems.

The continuous development of this system over a period of ten years revealed a number of important aspects that are crucial for the usefulness of a visualization system. Furthermore, some design choices underlying FAnToM are uncommon among visualization systems in general. Here, it is our aim to discuss some aspects and design choices underlying the FAnToM system to illustrate some of its properties and differences from other visualization systems. During the discussion, we will point out some experiences and lessons learned in working with the system on modern visualization applications.

## 2 Design Choices and Lessons Learned

The question of distinguishing the FAnToM visualization system from other, similar systems and tool-kits is one of the starting points of this report. In working with collaborators in the area of fluid dynamics research, we found that their data sets (obtained using CFD simulation) are usually given on large unstructured grids. Using FAnToM, they were able to quickly visualize data sets that were too large and too complicated for their commercial tools on existing hardware. The ability to handle large unstructured meshes with millions of cells was essential to them and enabled qualitatively superior visualization and analysis. We will describe next FAnToM's data structures and capabilities that facilitate the efficient handling of large unstructured meshes on commodity hardware.

*e-mail: wiebel@cbs.mpg.de
†e-mail: cgarth@ucdavis.edu
‡e-mail: hlawitschka@informatik.uni-leipzig.de
§e-mail: thomas.wischgoll@wright.edu
¶e-mail: scheuermann@informatik.uni-leipzig.de

[1] FAnToM – **F**ield **An**alysis using **To**pology **M**ethods

### 2.1 Data Handling

Unstructured meshes are supported in a small subset of visualization systems. The data handling modules of VTK [4] provide a solid and freely available implementation in the context of a powerful and versatile visualization toolkit. Initially, FAnToM's internal data structures were based on the corresponding VTK mechanisms. For performance reasons, however, it was found necessary to bypass the provided standard VTK interfaces and use internal data structures directly. Thus, parts of FAnToM relied on implementation details of VTK's data handler. Unavoidably, changes in the implementation within VTK eventually forced the implementation of a separate, custom data handling for FAnToM. Despite several reimplementation iterations that focused on increasing performance and memory efficiency, however, the general memory organization of unstructured data sets in FAnToM [1] is still similar to VTK's as described in the VTK book [4]. The small memory footprint of our data structures, i.e. the small overhead for organizing the vertices and cells, is the central feature underlying FAnToM's capability to treat large unstructured meshes.

### 2.2 Fast Integration-Based Visualization through Efficient Point Location

Many algorithms for visualizing vector fields, e.g. the computation of integral curves that is the basis for the integral surfaces, principally operate on continuous data and mandate interpolation of the discrete vector field under consideration. This requires, for each interpolation point, the determination of the cell containing the interpolation point. Since this is the innermost operation in all integration-based visualization algorithms, a fast method to perform this point location is crucial for the performance of visualization algorithms. Empirically, interpolating the discrete vector field dominates the computation time in this type of application.

For regular meshes, finding the cell containing a point is straightforward. As the (generally hexahedral) cells are aligned to the axis of the coordinate system, finding the correct cell reduces to comparing the coordinates of the position with the cell intervals along the axis. Point location in unstructured grids is much more complicated. Again, VTK provides a reliable point location method [4] using an octree-type data structure to subdivide the vector field domain. The tree leaves contain candidate cells that must be tested for point inclusion.

In working with unstructured meshes, we have found this approach to be unsuitable in the case of strongly non-uniformly distributed vertices, as often present in meshes resulting from adaptively refined simulation. The uniform subdivision of the employed octree tends to produce a significant memory overhead in these cases, while adapting poorly to the strongly varying cell sizes where it tends to waste memory [4].

The extensive work with such meshes performed using FAnToM quickly obviated the need for a different point location approach. The method developed by Langbein et al. [1] in 2003 specifically for use in FAnToM combines cell adjacency information and an adaptively subdivided *k*d-tree containing a small subset (typically

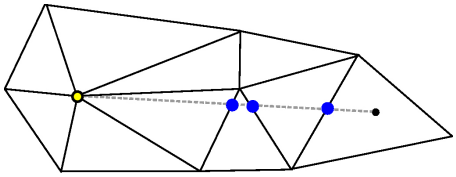

Figure 1: Casting a ray (gray) from the the vertex found in the kD-tree (yellow) to the sought position (black). At blue points the ray moves from one cell to the next using cell adjacency information.

1%) of the mesh vertices. The tree is used in a first step to identify a cell close to the interpolation point. From this cell, a ray is cast to the sought position using the cell adjacency for traversal (Figure 1). Special precautions are taken to reliably handle mesh holes and boundaries. The *k*d-tree and cell adjacency information can be build quickly when required and have a comparatively small memory footprint [1].

Overall, the up-front investment into the development of a custom, optimized data structure for unstructured meshes has had significant benefits. Besides enabling efficient application of existing visualization techniques, it has also facilitated the quick prototyping and development of novel techniques (such as e.g. [5]) that would otherwise have been prohibitively expensive to compute and develop.

### 2.3 Explicit Algorithm Execution

Many visualization systems are built on the premise of so-called *data flow networks*. These networks describe the flow of data from inputs such as data files through a sequence of filters to the output, which is typically a graphical representation of the original data. Data flow networks are very versatile, allow for parallel computation, and enable the construction of complex visualization algorithms from relatively simple component filters. Furthermore, such networks allow for caching of intermediate results in the case where only part of the filter sequence must be recomputed, e.g. in response to user feedback.

FAnToM uses a different approach that is aimed at explicit execution control by a visualization user. Using a custom plug-in architecture to support a large set of elementary algorithms, FAnToM allows for two different kinds of such algorithms: *Visualization algorithms* directly produce graphical representations from existing data, while *data algorithms* transform data sets. In this sense, visualization algorithms always present the last stage of a visualization technique, whereas data algorithms are intermediate pipeline stages. Execution or re-execution of these algorithms is explicitly controlled by the user, or implicitly performed through a scripting engine. Intermediate data sets are explicitly managed and can be written out to storage and re-loaded arbitrarily.

We have found this general philosophy of interacting with the visualization process beneficial in a number of specific settings. First, since FAnToM is aimed at treating comparatively large data sets on commodity hardware, the ability to explicitly save intermediate results has proven very valuable in performing visualization algorithms on such data. By effectively allowing the splitting of the pipeline at user-defined points, an expert user can thus carry out visualization tasks that would by far exceed the available memory in a data flow network setting. Secondly, we have found the additional flexibility of invoking arbitrary algorithms without the need to change the underlying data flow network valuable in increasing the interactivity of the visualization process. Lastly, during the development of new algorithms, this facility allows for quick sanity checks, thus reducing turn-around times and avoiding repeated recomputation of the same algorithms as required in a data flow setting.

### 2.4 Integration of New and Established Visualization Techniques

During our research in new visualization techniques, the work with our collaboration partners quickly showed us another precondition for the usefulness of a visualization system in general and the new techniques in particular: It is very important to have the newly developed visualization techniques together with well-established ones in a system. On the one hand, application domain users trust the methods they are familiar with, either because they know how the methods work (e.g. mathematically), or because they have seen the methods produce valid results often enough. On the other hand, many users distrust new methods; although, they might have many advantages, e.g. being more efficient and easier to use.

By presenting newly developed and well-established methods within the same environment, users are enabled to combine and thus compare the new and the well-established methods. As a result, it makes application domain scientists gain confidence in the correctness of the new methods. Using new and old methods together is also very sensible from a didactic point of view because the users will learn and remember more easily how to use the new methods if they applied them in a well-known context [3].

## 3 Results and Conclusion

Experiences from developing our visualization system over several years tell us that, if a system is aimed at more than a prototype, one should invest the effort in designing a customized fast and appropriate data handling that is tailored to the special properties of the data of the system's intended audience. Additionally, integrating common techniques into the software will greatly increase its usefulness.

Finally, it should be mentioned that FAnToM adheres to the aspects discussed in this abstract with its memory efficient data structures and algorithms and execution of the visualization pipeline. But it has some drawbacks as well, which are mainly regarding interactivity: FAnToM is quite flexible by its plug-in mechanism, but it is focused on autonomous processing algorithms that produce a visualization as output. These autonomous algorithms mostly only allow the specification of parameters in advance and do not provide means for changes after the execution of the algorithm. Elimination of this limitation will be the central part of the next large refactoring session for the FAnToM project in the future.


## Acknowledgements

The authors wish to thank the many developers for their valuable work on FAnToM over the years. Special thanks go to Markus Rütten, from DLR in Göttingen for years of fruitful discussions on fluid dynamics and CFD which ultimately also helped to improve FAnToM.


## References


[1] M. Langbein, G. Scheuermann, and X. Tricoche. An Efficient Point Location Method for Visualization in Large Unstructured Grids. In *Proceedings of Vision, Modeling, Visualization*, 2003.

[2] B. H. McCormick, T. A. DeFanti, and M. D. B. (eds.). Visualization in scientific computing. *Computer Graphics*, 21(6), November 1987. ACM SIGGRAPH, New York.

[3] J. Roschelle. Learning in interactive environments: Prior knowledge and new experience. In J. H. Falk and L. D. Dierking, editors, *Public institutions for personal learning: Establishing a research agenda*, pages 37–51. American Association of Museums, 1995.

[4] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit*. Prentice Hall, 2nd edition, 1998.

[5] X. Tricoche, C. Garth, G. Kindlmann, E. Deines, G. Scheuermann, M. Ruetten, and C. Hansen. Visualization of Intricate Flow Structures for Vortex Breakdown Analysis. In H. Rushmeier, G. Turk, and J. J. van Wijk, editors, *Proceedings of the IEEE Visualization 2004 (VIS'04)*, pages 187 – 194. IEEE Computer Society, October 2004.