

Exploring unsteady flows by parallel extraction of property-enhanced pathlines and interactive post-filtering

M. Vetter, S. Manten and S. Olbrich

Lehrstuhl für IT-Management, Heinrich-Heine-Universität Düsseldorf, Germany
{m.vetter,manten,olbrich}@uni-duesseldorf.de

Abstract

In this work a new approach of the visualization of unsteady high-resolution flow data in a network processing chain using property-enhanced traced particles or pathlines is presented. This approach allows to select subsets of pathlines according to additional given or calculated properties in the local environment and history of the pathlines and is an alternative method to classical feature extraction. As such, traditional property-controlled seeding strategies – as part of visualization mapping – are replaced by post-filtering based on multiplexed properties and geometries – as part of rendering. Inserted into our distributed visualization framework DSVR the selection of subsets is realized as an interactive “query over a stream” which considerably increases the degree of interaction in real time and also in 3D video-on-demand scenarios.

1. Introduction

Coming along with the compute power of modern super-computers actual numerical simulations can produce a huge amount of data. In typical approaches of visualization, the mapping of the raw data into 3D geometries and the rendering are done in a separate post-processing step after the simulation.

Techniques for flow field visualization can be classified into *direct flow visualization* like drawing arrows for each vector stored in the field, *texture-based flow visualization* like LIC, *geometric flow visualization*, which consists of particle tracing using numerical integration, and *feature-based flow visualization*, where the flow field is analyzed for so-called “critical points”. Overviews about these classifications and typical techniques are given by [PLV*02, LHD*04]. The visualization of special local or global characteristics of flows can be done by topology-based feature extraction techniques like in [TS03] or by methods based on vortex features as described in [TSW*05]. Another way is the filtering of textures or geometry based primitives according to additional properties. In [CFP00] the authors combined texture-based flow visualization with feature-based approaches. The extracted features are used to accelerate the time consuming texture-generation process. Recently an interactive feature-based filtering on attributed pathlines was

introduced in [STS07], but – differently to our approach – in a classical post-processing scenario with small sets of raw data.

Since numerical simulations of unsteady flows in high resolution take advantage of parallel high performance computing, it is very desirably to do so also for particle tracing and pathline extraction. The parallel simulation and the parallel visualization of particle traces which could render up to 60000 particles have been discussed in [BKHJ01]. In [SBK07] real-time flow visualization is done by analyzing regions of interest on a high-performance computer and do the flow visualization on a graphic frontend. [YWM07] introduces hierarchical representation in parallel visualization environments to increase scalability.

In these approaches the complete visualization is done as a separate post-processing tasks and all raw data have to be stored permanently. This may lead to a capacity bottleneck on the storage side. Considering a grid dimension of 10^{10} calculated over 10^4 time steps, up to 1 PByte would have to be stored in a computational fluid dynamics application.

To take these challenges a distributed visualization environment has been implemented in the *DSVR Framework* [OPR01, JOPR02, OMJ07] avoiding the storage of raw data at all.

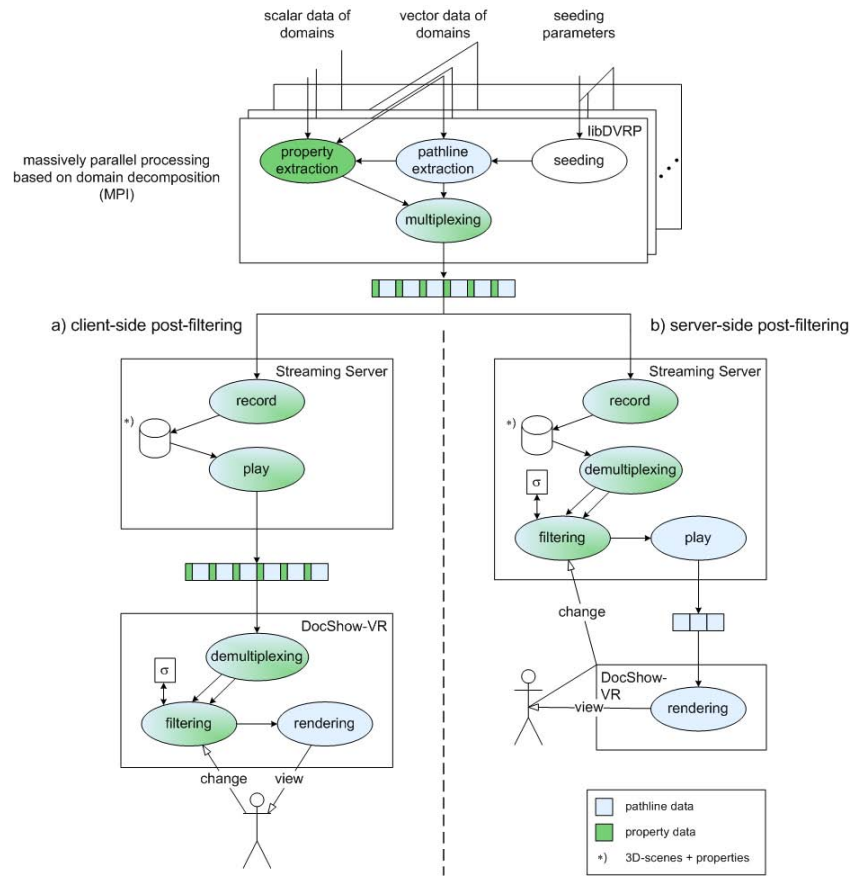


Figure 1: Process chain for massively parallel data extraction and highly interactive post-filtering visualization.

2. Concept and Design

A scalable approach including flexible support for batch, tracking, and computational steering scenarios is realized by our networked process chain, the Distributed Simulation and Virtual Reality Environment *DSVR*:

1. Data extraction and creation of 3D scenes, which represent features of the raw data, are efficiently implemented by parallel processing of the data parts – using a parallel software library *libDVRP* – corresponding to the domain decomposition of the parallelized simulation. This significantly reduces the data volume, while 3D interaction support is preserved.
2. The generated sequence of 3D files is stored on a separate *3D Streaming Server*, which provides RTSP-based play-out capabilities for continuous 3D media streams, especially in high-performance IP networks.
3. The 3D scene sequence is presented as an animation in a virtual reality environment. This step has been implemented as a web-based 3D viewer plug-in, taking advantage of stereoscopic displays and interactive tracking devices.

2.1. Parallel pathline and property extraction

The pathline extraction we have implemented as part of the *libDVRP* uses selectively the Euler or Runge-Kutta (2nd or 4th order) integration.

The global grid is distributed in partial grids over the cluster and the simulation's resulting data will be neither stored on disk nor communicated between the parallel cores. So the parallelization is ruled by the grid's domain decomposition and during integration of the trajectory, three cases for data locality in the compute cluster have to be considered.

1. The pathline will still stay in the grid volume, the actual process simulates.
2. The pathline will leave the whole simulated area.
3. The pathline will leave the area simulated by the actual process but still stay in the global simulated grid.

In the third case the pathlines are transferred between processes, so a pathline is always hosted by the process which also hosts the currently needed raw data. Using Runge-Kutta for integration, a transfer could also be necessary during a Runge-Kutta substep.

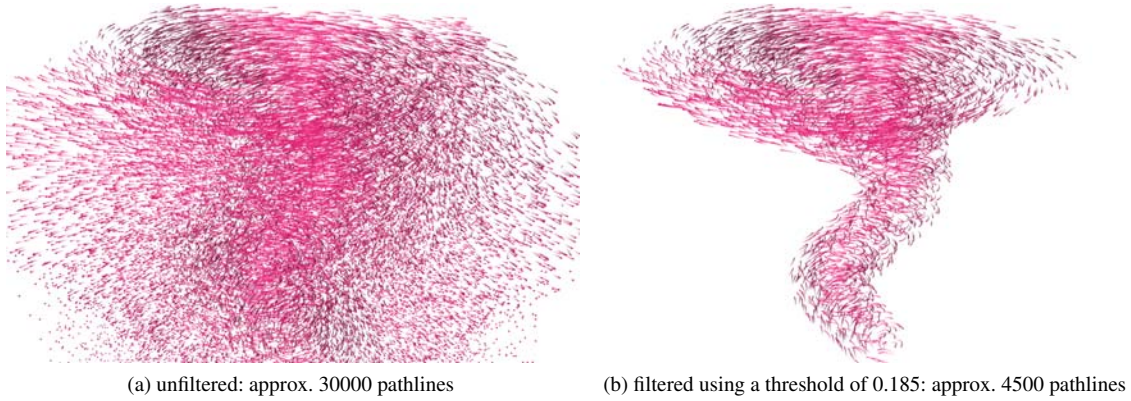


Figure 2: Flow visualization of a tornado-like simulated swirl using illuminated pathlines in *DSVR*.

The pathline extraction is implemented as on-the-fly processing, directly accessing the transient raw data of the respective time step in main memory, according to the underlying domain decomposition. As a result, we get an interleaved data stream, containing 3D geometry and additional property elements. Assuming a constant number of pathlines, $O(n^3)$ grid points and $O(n)$ supporting points per pathline, the data volume is reduced by a factor of $O(n^2)$.

2.2. Flexible, property-based pathline filtering

In graphics based approaches like streamlines or pathlines for flow field analysis, one of the most discussed problems is the initial placement of these lines. Seeding too many lines will lead to a unclear visualization. On the other hand, seeding only a few lines may result in missing some interesting features of the flow field. Usually this problem is solved by calculating adequate seedpoints.

But there is a problem about analyzing unsteady flows, because the interesting features of the flow field may change each timestep. Using typical post-processing applications, it is possible to first analyze the whole flow field over all timesteps and afterwards seed the lines according to this analysis. In addition to that, in highly interactive scenarios finding the best seedpoints may be an iterative process involving the user to adjust seeding parameters. But without storing the raw data like it is done in our approach, this prior analysis is not practicable. So another approach to solve the seedpoint-problem for simulation and visualization of unsteady flows on very large grids is needed.

Nearly the same degree of flexibility can be achieved by first storing a huge amount of homogeneously seeded pathlines which not only contain the actual position and history of the pathlines but also additional data generated by the simulation environment as a set of properties. For a clearer view and a better understanding of the flow features, the pathlines

will be filtered afterwards on the client side by changing type and parameters of a query function σ . This allows a high degree of interaction between the user and the visualization by controlling a filter function based on one or more selected properties. The properties are given by or derived from the simulation results at current trajectory position and optionally the pathline's history, as part of the pre-visualization task.

Figure 1 shows the data flow through the *DSVR Framework*. There are two approaches for handling the properties implemented in the *DSVR Framework*. In the first approach (a), the pathlines will be streamed through the *Streaming Server* without a change. Here the only purpose of that server is to store the 3D scenes and properties. The demultiplexing and filtering will be done by the *DocShow-VR*. In the second approach (b), the demultiplexing and filtering is done by the *Streaming Server*. This way only the filtered pathlines without property will be streamed to the *DocShow-VR*. Assuming a high-bandwidth network connection between the parallel cluster and the *Streaming Server*, transport and storage of a huge, but constant count of pathlines can be handled adequately. Streaming only the reduced pathlines to the *DocShow-VR* is the basis for applications in usual LAN/WAN scenarios.

3. Results and Discussion

We have implemented and evaluated our approach on a well-known routine by Crawfis[†] producing a tornado-like swirl on a 3D rectilinear grid[‡]. It was simulated on a grid with a dimension of 500^3 grid points each containing a vector (u, v, w) . As property we chose a scalar value defined by $\sqrt{u^2 + v^2 + w^2}$ at the current time step. Figure 2a shows a

[†] <http://www.cse.ohio-state.edu/~crawfis/Data/Tornado/tornadoSrc.c>

[‡] Project funding: DFG (GZ: OL 241/1-1)

points per pathline	unfiltered		threshold 0.209	
	data rate [MBit/s]	data volume [MByte]	data rate [MBit/s]	data volume [MByte]
40	510	181	60	21
20	260	92	31	11
10	134	47	16	5.5
5	70	25	8.2	2.9

Table 1: Network bandwidth streaming 5000 pathlines using server-side post-filtering.

Streaming Server: Intel Core2Duo 2.4 GHz, SuSe 10.2;
DocShow-VR: 2x Intel Xeon 5160 3.0 GHz, Windows XP,
 NVIDIA Quadro FX 5500; Gigabit-Ethernet connection;
 Dataset: 60 frames at 20 frames/s.

scene rendered in above discussed flow field with 30000 pathlines seeded. Here it is clear to see that seeding too many pathlines will lead to cluttered visualization where the swirl in the center is hard to see. After property-based filtering the swirl is clearly to see. Figure 2b shows the same view of the same scene as figure 2a but rendering a subset.

Furthermore this example shows the advantage to extract pathlines in areas where they are at first sight not really interesting. This could be seen when the swirl moves and particles, which were not part of the swirl at one time are points of interest at the next timestep.

The described testcase shows the opportunities of the here discussed approach as well as its estimated limits. Extracting 30000 pathlines using 40 supporting points and one property for each line results in 18 MByte per scene to store, load and stream. Aiming for 25 scenes per second, a network with a guaranteed bandwidth of 3.6 GBit per second is required. Based on 10-GBit/s-Ethernet this is already possible, but usually 10-GBit/s-Ethernet is not available in today's office networks. Using the data flow described in figure 1b, it is possible to store those data in adequate time and only stream the filtered 3D geometries.

Beside filtering whole pathlines we compress data by filtering supporting points for the pathlines. In first tests the network bandwidth needed for streaming 5000 pathlines was measured as shown in table 1. Due to the filtering on the *Streaming Server*, the reduction of streamed data volume is proportional to the reduction of supporting points per pathline by neglecting intermediate points.

References

[BKHJ01] BRUCKSCHEN R., KUESTER F., HAMANN B., JOY K. I.: Real-time out-of-core visualization of particle traces. In *PVG '01: Proceedings of the IEEE 2001 symposium on parallel and large-data visualization and graphics* (Piscataway, NJ, USA, 2001), IEEE Press, pp. 45–50.

[CFP00] CHEN L., FUJISHIRO I., PENG Q.: Fast lic image generation based on significance map. In *ISHPC '00: Proceedings of the Third International Symposium on High Performance Computing* (London, UK, 2000), Springer-Verlag, pp. 537–546.

[JOPR02] JENSEN N., OLBRICH S., PRALLE H., RAASCH S.: An efficient system for collaboration in tele-immersive environments. In *Proceedings of the Fourth Eurographics Workshop on Parallel Graphics and Visualization (EGPGV-02)* (New York, Sept. 9–10 2002), Spencer S. N., (Ed.), ACM Press, pp. 123–132.

[LHD*04] LARAMEE R., HAUSER H., DOLEISCH H., POST F., VROLIJK B., WEISKOPF D.: The state of the art in flow visualization: Dense and texture-based techniques, 2004.

[OMJ07] OLBRICH S., MANTEN S., JENSEN N.: Scalable isosurface extraction in a parallelized streaming framework for interactive simulation and visualization. In *Proceedings of the 10th International Conference on Humans and Computers (HC-2007)* (2007), pp. 147–152.

[OPR01] OLBRICH S., PRALLE H., RAASCH S.: Using streaming and parallelization techniques for 3D visualization in a high-performance computing and networking environment. In *HPCN (2001)*, Hertzberger L. O., Hoekstra A. G., Williams R., (Eds.), vol. 2110 of *Lecture Notes in Computer Science*, Springer, pp. 231–240.

[PLV*02] POST F., LARAMEE R., VROLIJK B., HAUSER H., DOLEISCH H.: Feature extraction and visualisation of flow fields, 2002.

[SBK07] SCHIRSKI M., BISCHOF C., KUHLEN T.: Interactive exploration of large data in hybrid visualization environments. In *Proceedings of the IPT-EGVE 2007 Symposium* (2007), pp. 69–76.

[STS07] SHI K., THEISEL H., SEIDEL H.-P.: Path line attributes - an information visualization approach to analyzing the dynamic behavior of 3d time-dependent flow fields. In *Topo-In-Vis 2007* (Grimma, Germany, 2007), Springer series of Mathematics and Visualization, Springer, pp. 60–74.

[TS03] THEISEL H., SEIDEL H.-P.: Feature flow fields. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 141–148.

[TSW*05] THEISEL H., SAHNER J., WEINKAUF T., HEGE H.-C., SEIDEL H.-P.: Extraction of parallel vector surfaces in 3D time-dependent fields and application to vortex core line tracking. In *Proc. IEEE Visualization 2005* (Minneapolis, U.S.A., October 2005), pp. 631–638.

[YWM07] YU H., WANG C., MA K.-L.: Parallel hierarchical visualization of large time-varying 3d vector fields. In *Proceedings of ACM/IEEE Supercomputing 2007 Conference* (Reno, November 2007), ACM/IEEE.