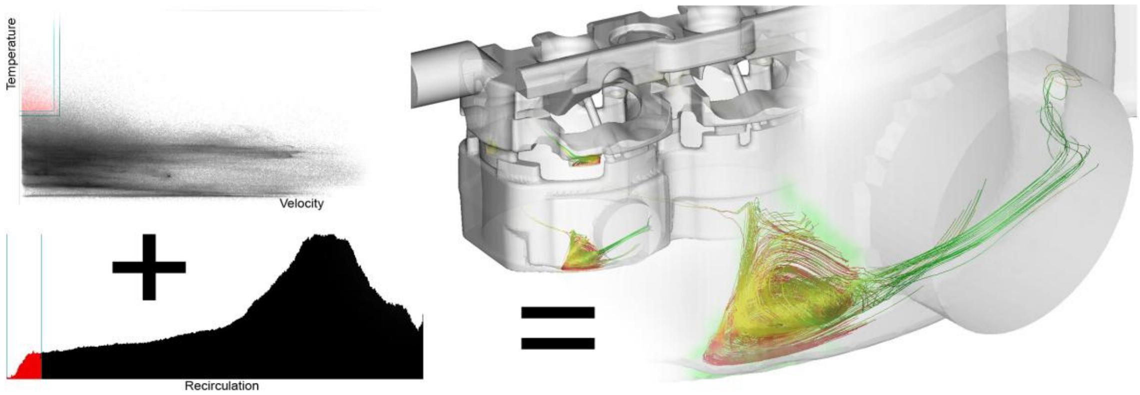
A Semi-Global Approach to Interactive Visual Analysis of Multivariate Flow Simulation Data

M. Ilčík
Faculty of Mathematics, Physics and Informatics
Comenius University, Bratislava, Slovakia
and
VRVis Research Center, Vienna, Austria

ilcik@cg.tuwien.ac.at



Contents

Abstract

We introduce a framework for interactive visualization of global flow features in large unsteady 3D flow fields. It is based on selective visualization using dense precomputed integral lines (streamlines, path lines) linked together with all other data attributes. This way we are able to provide an uniform and interactive environment for custom feature specification and visualization of non-local data aspects. These are related to the long term flow behavior, thus their description using only local properties is not possible. We demonstrate the benefits of working with semi-global features by a simple strategy for

recognition of recirculation zones within 3D vector fields, judging on a self-proximity measure of integral lines.

1 Introduction

Flow visualization deals with multivariate vector field data defined over a geometric mesh. Its application scope spans from automotive and aerospace industry through medicine right up to meteorology. Typically the data sets are produced by computational fluid dynamics simulations, containing over a million cells in several time steps, reaching the edge of present PCs processing capabilities. Thus, special visual analysis tools are necessary to understand the data as a whole. A good visualization of the flow field is very important for efficient creation of a mental model that helps the user to orientate in the data volume.

Given a 3D time-dependent vector field $\boldsymbol{v}(\boldsymbol{x},t)$, we can trace the movement of a massless particle from a position $\boldsymbol{x}_0$ at a time t_0 by solving the initial value problem for an ordinary differential equation

$$\frac{dx}{dt} = v(x,t), \quad x(t_0) = x_0 \quad (1)$$

Its trajectory can be represented by constructing a path line starting at position $\boldsymbol{x}_0$ at time t_0

$$p_{x_0,t_0}(t) = x_0 + \int_0^t v\left(p_{x_0,t_0}(\tau), \tau + t_0\right) d\tau \quad (2)$$

Another class of lines is often used to visualize the tangent curves of $\boldsymbol{v}$ for a fixed time t. A streamline starting at position $\boldsymbol{x}_0$ at time t_0 can be written as

$$s_{x_0,t_0}(t) = x_0 + \int_0^t v\left(s_{x_0,t_0}(\tau), t_0\right) d\tau \quad (3)$$

Hauser [9] gives a more detailed overview of integral curves. In order to solve the initial value problem and to construct the integral lines, we need to discretize the problem by rewriting the Eq. (1) to a discrete form. Using a substitution with finite steps $\Delta\boldsymbol{x}$ and t, we get

$$\Delta x \quad = \quad \Delta t \cdot v(x,t) \quad (4)$$

Once a step length $\tau = \Delta t$ is chosen, one can express the particle's position after $n + 1$ steps using a simple recurrence called Euler integration method

$$x_{n+1} = x_n + \tau \cdot v(x_n, t_n) \quad (5)$$

For a more accurate numerical integration, higher-order methods from the Runge-Kutta family with a fixed or adaptive step size can be used. The most popular are RK4, RK3(2) and the Fehlberg method, all explained in [20].

Integral lines belong to the group of geometric flow visualization techniques, because they require to extract geometric objects related to the data in order to display them. Our approach concentrates on feature based visualization. Especially for large data sets it is important to allow a custom degree of abstraction showing only essential structural elements of the flow. In our framework we provide a set of tools allowing the user to specify interesting features using selections on various attributes. Data values, mesh connectivity and integral lines act as layers for our concept of uniform manipulation with local and global properties. By linking these layers together, we can significantly help to gain insight into a specific problem represented by the data.

Using our approach, it is possible to simultaneously analyze the multivariate simulation data while at the same time learning more about the related flow behavior. The Eulerian and Lagrangian points of view are presented together, so that the analyst can track flow features along the flow and examine their interaction with local data characteristics at once. The scale of the flow structures can vary from very large to microscopic small. Understanding relations between data attributes and the underlying flow behavior, especially the interaction of large scale and small scale factors, is very important for successful problem solution. Rapid interactive flow exploration following this concept is so far unique to our approach, along with its support of unstructured non regular meshes.

The rest of this paper is organized as follows: Section 2 provides a brief overview of similar or related work to our concept. In section 3, we define the concept of a layered model for visual analysis of multivariate flow data. Moreover our contribution shows how to unify and link all these layers to an uniform environment. Section 4 presents our framework in detail. We propose a simple recirculation measure based on global flow analysis in section 5. A practical application of the layered model to industry data analysis is described in section 6. Finally we summarize our results and consider possible future work in section 7.

2 Related work

Utilizing particle traces for visualization of fluids has always been popular. Texture based visualization uses local integration for excellent interactive renderings of static and dynamic flow [12]. Feature based analysis and visualization of flow structures with help of enormous amounts of integral lines has gained attention only recently. Park et al. propose a visualization technique based on dense, uniformly distributed integral lines [18]. They address many important perceptual and performance issues. Chen et al. use a similarity measure of polylines for streamline placement [2]. Wischgoll and Scheuermann present a strategy for localization of closed lines in 3D [24] using the mesh topology to track line cycles reentering the same cells. A more general approach by Salzbrunn and Scheuermann [22] is heading in the direction of custom feature specification. It uses different data measures for the specification and computation of fuzzy streamline predicates, similar as we do in our global scope flow analysis approach. The selective visualization part of our framework and its usage for line data analysis comes very close to the approach by Shi et al. [23]. The authors present a set of path line measures suited for general flow analysis. For interactive examination of lines they use ComVis, a system oriented to information visualization. Unfortunately another system has to be used for computation and visualization of lines on the geometrical domain, preventing general interactivity, usage freedom, attributes linking and perception of spatial relations.

Detection of recirculation areas addressed in section 5 is related to the vortex detection topic, well described in the state of the art report by Post et al. [19] and also in context of SimVis by Bürger et al. [1]. Fuchs et al. [7] use parts of our framework for an improvement of vortex region detectors. They track the response to local vortex detectors along integral lines and use it for a global scope flow analysis.

The SimVis system

Our work presented in this article was implemented as a part of the SimVis system [4], a modern visualization framework aimed at interactive exploration and analysis of multivariate flow data. It allows the user to formulate own definitions of features describing significant objects, phenomena or structures, important for a particular problem. Each data item has a degree of interest (DOI) value ranging from 0 to 1. It is used to reveal the implicit data features by specifying selections on their values. To simplify the selection process of continuous quantities, smooth brushing is used [5]. For the DOI management, there is a hierarchical tree structure called feature definition language (FDL). DOI functions are stored in leafs, any other node represents a logic combination of its children [4, 6]. SimVis uses multiple linked views to simultaneously provide different visualizations of various data properties of the data. Active views utilize InfoVis approaches and allow brushing, passive views show a 3D visualization on the geometrical mesh with the FDL root node mapped on the data. The views are linked together with a joint DOI function, so any user action causes changes in all of them. The possibility of data exploration in several linked views at once allows to investigate complex relationships between the scalar attributes and the underlying vector field, using various interpretations and abstractions. SimVis is based on the Focus+Context visualization approach[8], thus all views have to use the DOI to discern focus from context. Focused regions are visualized with more emphasis and detail, context parts provide only supplemental overview. An example of the SimVis user interface is provided on Figure 1.

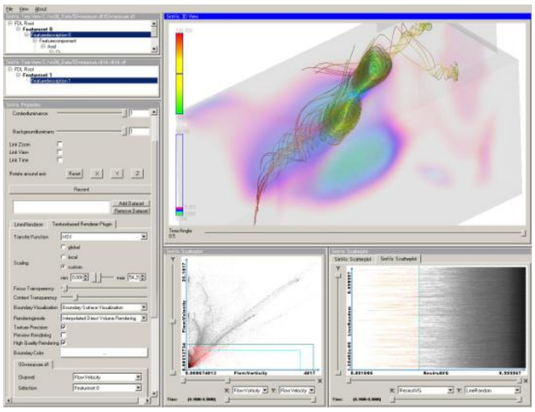


Figure 1: A typical SimVis layout.

In the next sections we will exploit the concept of SimVis in order to allow global scope flow analysis.

3 Flow driven analysis

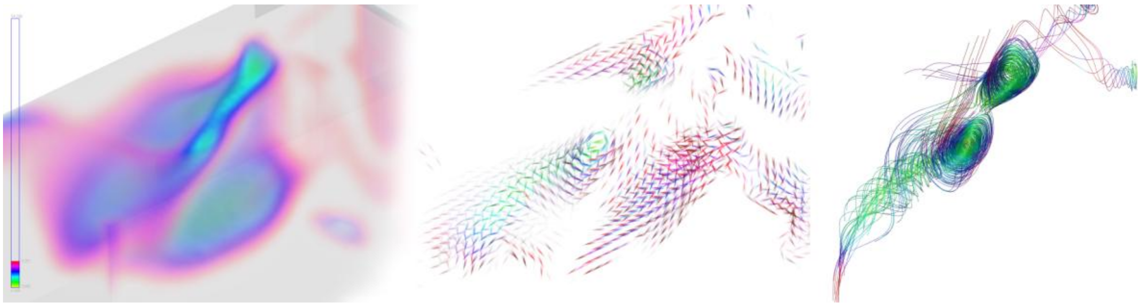


Figure 2: A comparison of visual analysis layers. Color represents velocity magnitude. *Top*: Regions with low velocity magnitude are displayed using raycasting of the direct layer. *Middle*: Using the connectivity layer information, cells with strong response to λ_2 vortex detector are displayed with arrows pointing in the velocity direction. *Bottom*: Recirculating lines of the integration layer reveal two vortical structures with slow vortex cores.

The essential asset of our framework is its ability to provide a flexible environment for coordinated interactive visual analysis of global data aspects. In this article, we present an extension of the SimVis system with a new integration based analysis layer that provides the user with means to investigate the long-term flow behavior related to the specified features. Our main goal is to allow the flow analysis from a Lagrangian point of view and to connect it with other common techniques — mainly with those using the Eulerian point of view, and with various statistical methods.

The Eulerian point of view is considered to be spatially local, since the flow is described as a vector field reflecting changes of particle positions at a fixed geometrical point. It is well suited for local scope analysis of single quantities, but we need to use a different approach for a semi-global analysis of the data. The Lagrangian point of view represents the flow by particle trajectories, allowing to investigate changes occurring along the particle path, or relative to other particles. In order to be able to use it for visual data analysis, we need to transform the representation of the vector field to a set of trajectories. We use large amounts of precomputed integral lines as primitives for the global data analysis. The trajectories gained by following one or more particles in the vector field can span over a large portion of the spatio-temporal data domain, representing global properties of the flow. The line primitives provide us a new kind of complex structural information that could never be obtained without the integration process. Many features like recirculation (see section 5) are imperceptible without a global scope analysis. This fact leads us to a clear separation of three layered approaches for flow data visual analysis used in our framework. All described layers can access the cell centered data storage and inferior layers for purposes of their own primitives computation. A comparison of renderings at different layers is provided on Figure 2, a scheme of the layers on the Figure 3.

Direct data analysis layer

> is restricted to the raw data values, without involving any geometrical knowledge of the underlying mesh. We can consider a data value of a cell as the basic entity for this layer. The minimal amount of information used, allows fast processing even for large data volumes. During the analysis it typically uses only a set of interval brushes in data attribute dimensions for selecting cells with certain interesting values like low pressure, high temperature etc. Indirectly, a kind of connectivity can be observed in this layer as well. In the case of a continuous function, selected intervals contain a range of continuous data values that should be affiliated in neighboring cells.

However, generally we can not assume a direct geometrical relation between selected cells. This independence from the mesh topology represents the strongest feature of this layer, making it the only one capable of filtering and manipulating certain cells out from their geometrical context.

Connectivity based analysis layer

offers additional connectivity information involving the topological structure of the mesh. The basic entity on this stage is a set of neighboring cells, suitable for local processing like boundary detection. In connection with the previous layer, it enables the usage of differential information for evaluation of local data properties. The flow is being observed from an Eulerian point of view, examining the behavior of particles streaming through a point and its surroundings. Many well established local feature detection criteria are based on spatial or temporal differentials, e.g. different vortex detectors using the velocity gradient tensor [1, 2]. Selections on this layer brush cells according to the attributes derived from the grid geometry and connectivity, like the already mentioned boundary detection. However there are global measures based on this connectivity based analysis layer as well.

Integration based analysis layer

extends the connectivity information to a wide area, in our case restricted to the integral line seeded at the examined point. This layer exploits the Lagrangian approach, examining the set of positions a single particle streams through. Therefore it is the best to use a set of connected segments forming an integral line as the basic entity for this layer. The main benefit for the exploration process is the possibility to investigate the long-term flow behavior before it reaches and after it leaves the examined region. Already for the computation of integral lines, information from both superior layers is necessary. The data layer provides the velocity vector that is further used by the connectivity layer to determine the cell for the next integration step. Furthermore it allows analysis based on local attributes of line points and global line properties. Many recent works use global scope flow analysis, e.g. [22, 23]. However we can not characterize this layer as being strictly global, since we often use also a local analysis of single line segments quantities and show rather all segments crossing a cell then a whole line seeded there.

Table 1: Layers for interactive flow analysis.

Layer	Primitive	Source	Display
Direct	value	data channel	color coding
Connectivity	set of cells	mesh topology	arrow glyphs
Integration	integral line	particle paths	lines

Most of the present systems for interactive visual analysis of 3D flow data can handle only separate layers at once. There is no possibility of data exchange between different primitives. In our framework we introduce the interactive linking and brushing [5] of these three layers. The linking involves not only the access of the superior layers to the lower ones, but mainly the automatic bidirectional mapping of the DOI values between the layers. The cells act as an interconnection of the analysis primitives, thus a bijective transformation between the primitives of each type and the cell set is available. Using our approach, it is possible to combine selections on all three layers to a uniform DOI rating of the cells. This increases the expressive power of the custom feature specification, allowing descriptions of flow structures based on long-term particle behavior in the stream. In order to preserve uniformity, simple and intuitive handling, we try to avoid separate selection mechanisms for each layer. Most of the specific selection techniques from higher layers can be reformulated as simple measures implicitly encoding the connectivity and trajectory information into raw data values, which can be brushed using the direct data layer. Technical details of the layers linking mechanism are explained later in the section about interactive lines manipulation.

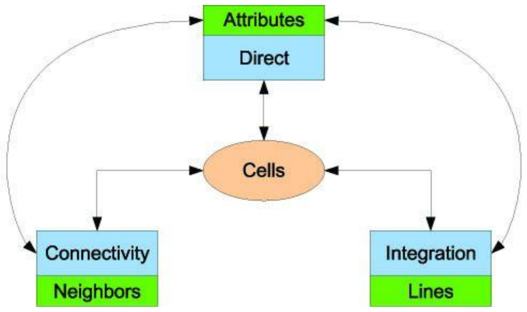


Figure 3: Layers linking. Cells connect the layers by combining the DOIs. Properties of superior layers can be reformulated as simple data attributes and brushed using the direct layers.

4 Framework overview

Industrial flow data sets produced by complicated computational fluid dynamics simulations contain several millions of cells stored in one or more time steps. The resolution of the underlying grid is adaptively adjusted in order to maintain a given degree of precision. It can differ in several orders of magnitude across the data set. Thus, the scale of the flow structures can vary from very large to microscopic small. Understanding relations between data attributes and flow structures, especially the interaction of large scale and small scale factors, is very important. Our approach allows custom combination of coordinated multiple views [21] and visualization techniques, in order to enable a direct analysis of relations between raw data values, local and global flow properties.

The set of proposed methods was designed and implemented as a part of the SimVis system, hence it closely relates to its concept and architecture. The direct data analysis layer is a core part of the system. We have incorporated the local scope analysis in the last years [1]. For compatibility reasons, new methods need to be implemented for unstructured non-regular meshes. Since the integration based layer is the only one to require additional offline precomputation in order to maintain interactivity of the exploration process, we have designed two modules operating independently from each other. We can divide a typical usage scenario into several stages, realized by the modules as seen on the Figure 4

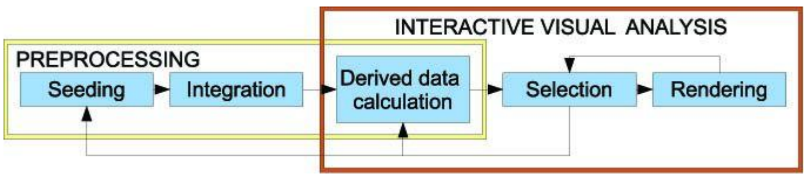


Figure 4: Pipeline for our framework.

Preprocessing module

We use large amounts of precomputed integral lines as primitives for the global flow analysis. In order to create them, we need to set seed points or a seeding strategy and integrate from this positions using a continuous representation of the vector field. Setting the seed point at cell center is a good choice for the later linking, since the data values in our application are also sampled at cell centers. By default we create one representative line for each cell, but the user can decide to omit computation of unimportant lines by seeding only in specified cells with positive DOI.

For the numerical integration one has to choose between speed and accuracy. The fastest one is a direct Euler method with step size adapted to the underlying geometry. It is using nearest neighbor lookups for the velocity vectors. Error rate of this integration method lies above the tolerance threshold in most of the cases. For a higher precision one can choose some of the adaptive Runge-Kutta methods with mean value coordinates interpolation [11]. All lines can be limited by a maximum length setting. Currently we offer two types of lines to be computed: streamlines and path lines. The lines are saved in the form of a new data set. Its size can be estimated by the upper boundary $cells_{selected}$ * maxsteps * 26 bytes, hence it can easily reach problematic limits of several GB.

Realization of our solution has shown to be really complicated and challenging. Fast and precise numerical integration on unstructured 4D grids is not straight forward. Hence it requires to employ special solutions trying to reutilize as much information as possible. We have developed some grid aware methods that reduce the lookup time when searching for a cell where the integration scheme will make the next velocity vector lookup. The lines are seeded in cell centers, thus we already know the starting cell. When the particle trajectory leaves the current cell volume, it is easy to determine the exit face and the cell on the other side. One integration step may cross an arbitrary number of cells, so compared to a spatial hierarchy search that finds directly the last cell in O(log n) time, our approach needs O(n) time in worst case. However, the step size is because of precision reasons always held much smaller than the cell diameter, so the connectivity usage improves the lookups in average case to constant time O(1).

Additional derived data, mostly local and global measures operating on the lines geometry, can be also precomputed along with the integration or later during the exploration process. We will provide an example and discuss benefits of this feature later in section 6 as we will use it for an estimation of particles density corssing a cell.

Table 2: Parameters stored for a line segment.

Channel	Start point	Time	Cell	Step	Line	Extra measures
Data Type	float vector	float	int	short	int	float
bytes	3*4	4	4	2	4	n*4

Interactive lines manipulation

The interactive part of the pipeline allows the user as much freedom as possible to organize and guide the analysis. Following the principles of knowledge driven visualization, the SimVis system provides the experts with means to formulate their own feature definitions [4]. The main part of the interaction relates to various selections on active linked InfoVis views [5]. Storing the integral data as a common data set allows us to reuse the core modules of the SimVis and to exploit its selective visualization capabilities. It allows to treat the segments as if they were regular cells, so the interface stays the same, only the data meaning is different. From the analysis layers point of view, the user can brush on the integration based level just like on the other layers. Hence any active view can be used to specify the FDL tree for the lines data. It is made possible by the linkage of cells and layer primitives (see Figure 3). Lines are build up of segments, connecting pairs of succeeding points. During the integration we split line segments at cell boundaries and time step borders, so that we can easily construct a bidirectional mapping of segments to cells. Moreover, each segment carries a reference to the seed segment of the line, allowing a bijective mapping from lines to cells. The segment to cell part of the mapping information is stored together with the line data as a segment attribute, as it contains only one entry for each segment. The cell to segment mapping part is a 1:n relation that needs to be attached to the original data. We store it with help of supplemental "interface" files. These are created right after the integration process and store two mappings:

- *Cell to segment* lists all segments within a cell.
- *Cell to line* lists all segments of the line seeded in a cell.

The DOI transfer between the original data set and the lines data set can be established by loading and activating one of these interface files. It is frequently used during the visual analysis, mainly for lines filtering described in the next section. The lines DOI can be also transferred back to the cells and used for example by the preprocessing module as the seeding area for a new set of lines. Transfer loops are leading to a deadlock and are therefore disallowed.

The manipulation with lines is not limited only to brushing and selections. It is also possible to supplement the existing lines with additional attributes. Many of the various derived data calculations provided by the SimVis core are working on lines as well. The data can be shared across all the layers, so it can be involved into different calculations. Various line measures — already mentioned in the previous section — can be also computed as derived attributes during the interactive analysis.

Visualization module

The most important type of view for the spatial orientation is the 3D renderer. It provides hybrid visualization of the data volume using multiple techniques simultaneously. Our lines rendering plugin can operate as a standalone renderer or as a part of a combined rendering pipeline [17]. The principal task of visualization bound to a geometrical domain is to provide a clear overview of the flow volume and spatial relations between the flow features. Large amounts of lines as we produce them, can easily overwhelm the image space. Intersections and cluttering cause problems by depth perception and orientation, important structures can be hidden. A lot of research has been done in this field: lines density regulation by advanced seeding strategies [16, 14] and rendering enhancements for better visual separation in dense line fields [18]. Our renderer uses illuminated lines [15] that greatly help to improve the recognition of single lines in the image. Naturally we can map scalar attributes to the color channel, using build-in transfer functions. The lines density is implicitly adjusted by the data selections. Automatic linking of the DOI across the views immediately propagates its values from the cells to their line primitives, so that only curves belonging to the brushed cells are shown in the final image.

Framework performance

The visualization module runs fully interactively using common personal computer hardware. The number of displayed lines is limited by the graphics memory and fill rate, but it rarely happens that the user would select such a big number of line segments. An optional random filtering of lines is always present as a fallback option reducing the rendering load.

The preprocessing module uses numerical integration methods to compute the lines. Computation time depends on the selected integration and interpolation method. On the fly integration would not allow interactive analysis anymore. We have measured the average performance of our integration methods on unstructured meshes using an AMD Athlon 64 3200+ processor with 3GB of RAM:

Table 3: Performance overview.

Method	Interpolation	Segs / s	Cooling jacket
Euler	none	$\approx 5.10^4$	≈25 min
RK2	mean value	$\approx 2.10^4$	≈68 min

5 Recirculation detection in steady flows

As stated in the survey of feature-based visualization [19], one of the areas with additional work needed is detection of new feature types, including recirculation zones. A lot of feature detection is done on a local basis, involving only a small geometrical neighborhood of investigated position. However, it is impossible to detect a recirculation area using only local detectors. Our precomputed lines provide a global representation of the flow field, containing new connectivity information suitable for efficient feature detection. If a particle advected in the flow field returns closely to one of its previous positions, we speak of recirculation. Streamlines contain all positions of a particle moving in a steady flow, making them ideal for a fast and simple detection of recirculation.

Given a line as an ordered sequence of points $c = \boldsymbol{x}_0, \ldots, \boldsymbol{x}_{n-1}$, we can precompute the length of the polyline $\boldsymbol{x}_i, \ldots, \boldsymbol{x}_j$ for all pairs of points i and j as

$$d_{ij} = d_{ji} = \sum_{i \le k < j} \|x_k - x_{k+1}\| \quad (6)$$

Due to the triangle inequality, the polyline length is always equal or larger than the Euclidean distance of its endpoints.

$$\forall i \neq j : d_{ij} \geq \left\|x_i - x_j\right\| \quad (7)$$

We define our recirculation measure $\boldsymbol{p}_i$ for point i as

$$p_i = \min_{j \in \{0, \ldots n-1\} \setminus \{i\}} \left(\frac{\left\|x_i - x_j\right\|}{d_{ij}} \right) \quad (8)$$

Since we deal only with non-degenerate segments, $d_{ij} > 0$ always holds and $\boldsymbol{p}_i \in (0,1]$. Let us consider a particle at position $\boldsymbol{x}_i$, moving in a vector field. As the particle moves forward passing through the points$\boldsymbol{x}_j$, the polyline length d_{ij} of its trajectory will grow. But when the particle starts to draw near to its own line, the Euclidean distance of the points $\| \boldsymbol{x}_j - \boldsymbol{x}_i \|$ will decrease. The response will grow weaker with increasing polyline length and decreasing Euclidean distance.

6 Application example

We will demonstrate the possibilities of our framework by resuming the analysis of a cooling jacket from [2, 10, 13]. Engineers from the automotive industry use computational fluid dynamics software to simulate fluid or gas behavior in possible designs of essential engine parts, and to examine their performance under various operational conditions. The SimVis system provides them an interactive visual analysis environment for investigation, comparison and optimization of the simulation results. The subject for this application study is a cooling jacket of a four cylinder car engine, provided by our industrial partner AVL List GmbH from Graz, Austria. The 3D steady flow data set is sampled in an unstructured grid consisting of 1.5 million polyhedral cells. A cooling jacket is build up of three main components: The cylinder head on the top, the cylinder block on the bottom and a thin gasket connecting the previous two components together (see Figure 5).

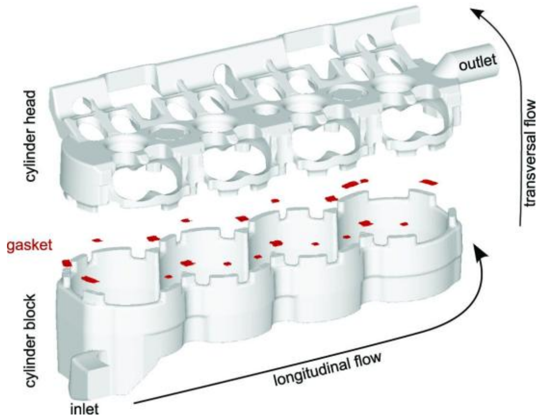


Figure 5: Cooling jacket components and optimal liquid flow directions.

The cylinder head
: is the upper part of the jacket, responsible for transferring the heat away from the intake and exhaust ports at the top of the engine block.

The cylinder block
: transfers the heat away from the engine cylinders and distributes the flow evenly to the head.

The gasket
: consists of small but important ducts regulating the passage of the cooling liquid from the block to the head.

There are two main components of the flow through the cooling jacket: a longitudinal motion lengthwise the geometry from the inlet towards the outlet and a transversal motion from the cylinder block to the head and from the intake to the exhaust side of each cylinder. The engineers would like to keep the engine operating at an optimal temperature level. A very large jacket would effectively transport the heat away, but it would shorten the space for other mechanical parts and also increase the vehicle weight. Compact geometry design results into complex shapes determined by the form of the engine block and by the desired fluid circulation inside of the jacket geometry. The most important design goals are:

- to achieve an even distribution of the flow to each engine cylinder
- to avoid overheating caused by slow transportation of the hot fluid

The first goal is rather intuitive, an even distribution of the cooling liquid to each of the engine parts implies even heat transfer form the cylinders, intake and exhaust ports. The second one is critical, since stagnant flow leads the absorbed heat not away. The engine does not cool fast enough, the accumulated heat can potentially lead to boiling conditions. An optimal operating temperature of the fluid is about $363K \approx 90°C$.

Both major design goals were directly or indirectly influenced by the fluid mechanics in the surroundings of the gasket. Thus the main variable for a cooling jacket design lies in the gasket. Engineers adjust the number, location, and size of the conduits in their pursuit of the ideal fluid motion.

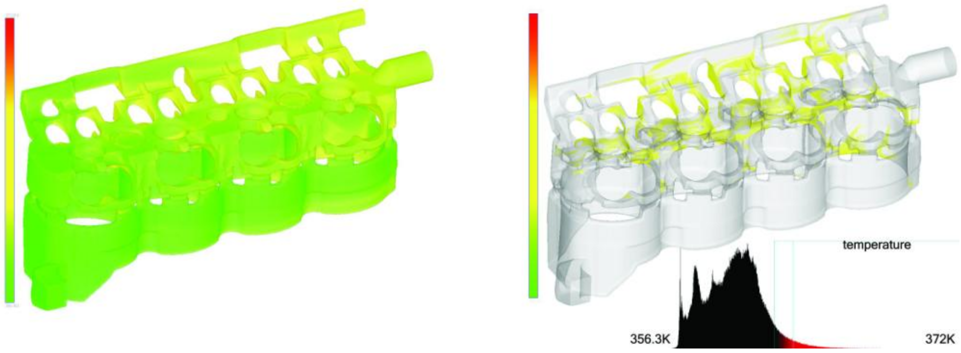


Figure 6: *Left:* Temperature in the jacket is increasing further from the inlet. It is difficult to find critical regions among all the cells. *Right*: A selection of temperatures above the optimum provides a better, but still cluttered visualization.

Overheating areas

The most straight forward way to get overview of problematic areas is a direct mapping of temperature to color. This kind of visualization mapping requires a manual inspection of a large area that contains a lot of dispensable information. Such analysis may be tedious and error prone, since the recognition of overheated spots is left to the human visual system. Therefore it is better to use feature based visualization that helps to focus only on overheated regions. Using a threshold of 363K to display only cells with higher temperature than the optimum will reduce the amount of irrelevant information to minimum. A histogram provides additional information about the temperature distribution serving as visual help for defining the selection. You can see both renderings on the Figure 6. Although it is easy to identify the overheating parts of the cooling jacket, there is no evidence for the reason. We will need to investigate not only other physical attributes, but also the preceding flow behavior. Without a clear visualization of relations between the attributes and flow structures, it is very hard to judge about design deficits.

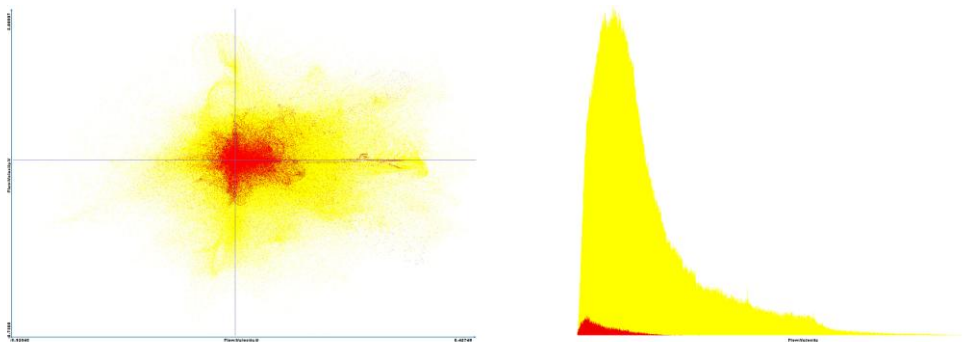


Figure 7: Temperature and velocity distribution. *Left*: Longitudinal and transversal flow mapped to a scatterplot. Entries with temperature over $363K \approx 90°C$ are colored red. There is no indication of coherence between recurring flow and overheating. However most of the problematic cells seem have low velocity magnitude in common. *Right*: Velocity magnitude histogram proving the previous statement with high temperatures shown with red.

Slow heat transport

Unilateral flow is preferred to recirculating flow, since it transports the cooling fluid more effectively away. Longitudinal and transversal flow represent the principal direction components, opposite direction may indicate non-optimal behavior. On the other hand, slow stream also leads to reduced heat transport. For a comparison of these two statements we use a scatterplot linked with a temperature selection (Figure 7). It shows the density distribution for all possible pairs of longitudinal and transversal velocity components combinations (meaning the X and Z components of the velocity vector). The cells selected as described in the previous section are displayed with a different color hue. With help of this InfoVis view, we can easily see that there is no correlation between returning flow and overheating at all. However it reveals a weak relation between slow cooling liquid transport and overheating. Based on this observation we can refine the selection to get a less cluttered overview of the most critical spots with high temperature and low velocity magnitude (Figure 8). Regions with increased temperature can be found on many places in the cylinder head, mainly at the second and third cylinder. The most critical areas are thin coves and bends in the upper part placed out of the way of the main fluid stream. There are also overheating rings just behind the main gasket ducts, probably caused by insufficient fluid spread. The only large non-optimal area in the cylinder block is located at the fourth cylinder.

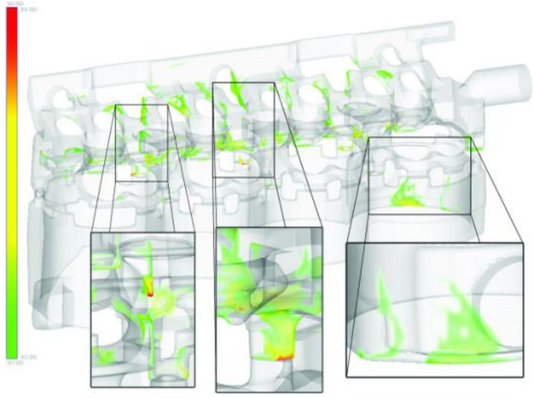


Figure 8: Stagnant, overheating regions. Overheating areas with stagnant flow. The color coding of temperature has been scaled to $[363K, 368K]$. Detailed images show instances of problematic parts as viewed from the opposite side of the jacket. We will investigate investigate these in the following sections. *Left*: Overheating in a thin cove. *Center*: Insufficient fluid spread behind the gasket. *Right*: The only problematic area in the cylinder block.

We have already examined the cylinder block in our previous work [2] using a typical workflow of direct and local flow analysis of velocity gradient tensors for detection of swirling motion. Indeed a vortex causing strong fluid recirculation and slow down in this area is responsible for the inefficient heat transport. Compared to other vortices in the jacket, this one is extraordinary slow, rotating with a velocity lower than 0.1 *m/s*. Since our previous publication, we have incorporated the integration based analysis layer in order to generate Runge-Kutta streamlines for the data. We can simply reconstruct the previous analysis by computing the $lambda_2$ criterion and selecting cells with negative values. This selection can be transferred to the lines data by the linking mechanism, so that only curves crossing the vortex cells become visible. In the following examples we rather use a semi-global recirculation criterion analyzing the self-proximity of lines, instead of the $lambda_2$, to select swirling structures. Compared to the previous results, the visualization using integral lines provides much better insight in the flow structure responsible for the increased temperature in this region (Figure 9).

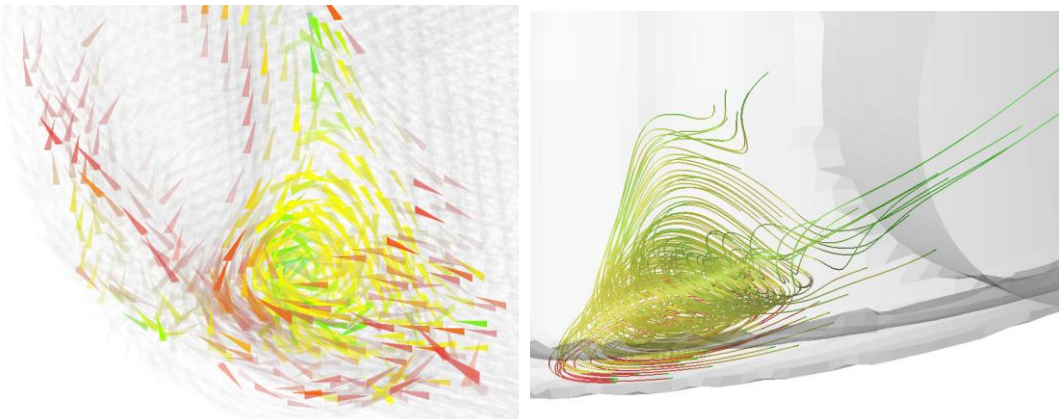


Figure 9: $Lambda_2$ analysis. *Left*: The $lambda_2$ analysis showing cells with $L_2 < -10^3$, using arrow glyphs pointing in velocity direction. *Right*: Our enhanced visualization with help of integral lines selected by the recirculation criterion. The color represents temperature scaled as on Figure 8.

In the cylinder head, we first look at the thin vertical coves at the exhaust ports. We have picked up the region of the first cylinder, but a very similar overheating can be found at all other cylinders. An inspection of the velocity magnitude shows the reason — velocity in the outlying corner is lower than 0.002 *m/s*. This nearly stagnant flow is far too slow to transport the heat efficiently away. Physical attributes give no further evidence for the reason of the low velocity magnitude in the corner. Hence, we need to analyze the flow structure in this region. Using a geometrical selection, we can isolate the lines seeded in the area. The local behavior of the cooling liquid is normal, no turbulences or other anomalies are present. But when we extend our scope a few centimeters towards the exhaust port, we will find a vortex that should spread the fluid to the surrounding coves. Probably there are not enough particles bounced from the turbulence to the corner, so the fluid drive gets weakened. We can verify this hypothesis using a simulation run of the streamlines integration. Thereby the computed lines are not stored as new geometry, they only provide a particle trace for semi-global analysis. We use the line density measure to estimate the particle flow rate through a cell.Figure 10 shows the results of this complex analysis. The strong vortex leads most of the fluid towards the upper part, only a small portion slowly flows into the cove. The extraordinary hot corner with temperature of 372*K* (nearly at the boiling point) has indeed only a weak stream density. The reason is definitely the nearby vortex, thus we can conclude that the geometry in this region should be improved in order to achieve a better spread of the cooling liquid to the outlying parts of the jacket as well.

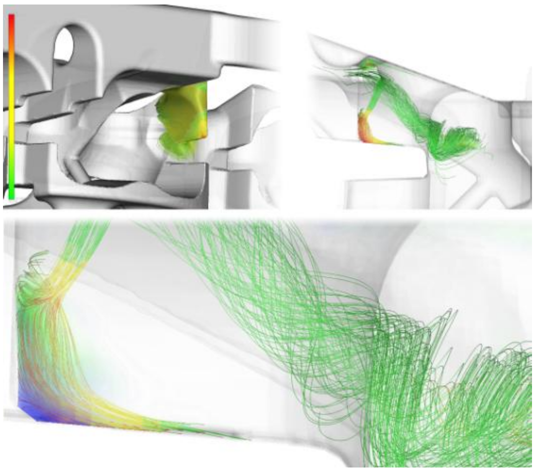


Figure 10: Overheating due to low spread. *Upper left*: Overview of the cove with temperature mapped to lines color. There is no filtering applied. *Upper right*: A side view of the cove. Using a union of two feature definitions it shows lines with high recirculation response and lines crossing the overheated left corner. Velocity is mapped to the lines color from red to green, showing the extreme slow liquid motion in the cove with green. *Down*: A detail of the cove with lines as on the upper right image. Additional volumetric rendering of particles density shows the sparse region coded with red.

The second region of common overheating present at all cylinders is a small ring right above main gasket ducts. We assume, that the reason will be similar as in the previous case. Again, we analyze the streamlines seeded in these cells and check the velocity and line density in this area. The thin gasket ducts accelerate the fluid so that the strong main stream flows upwards with almost no spreading to the side. Visualization of the fast inner flow structures is provided on Figure 12. There is enough free space between the hull and the main stream from the gasket, that needs to be filled. After a short investigation of the streamlines, we suspect that the fluid is coming from the upper region to fill this area. Figure 11 confirms such flow behavior. We have selected only lines with at least one segment pointing down (towards the negative Z axis) and mapped the step number to the color. This way, we can clearly see where a line begins and where it ends. The fluid is entering the ring around the fast gasket stream only from one side and leading through the opposite side. However as the Figure 11 shows, most of the fluid particles do not reach the other side of the ring, but rather join the stream in center. Thus the heat transfer is constrained and the hull is overheating.

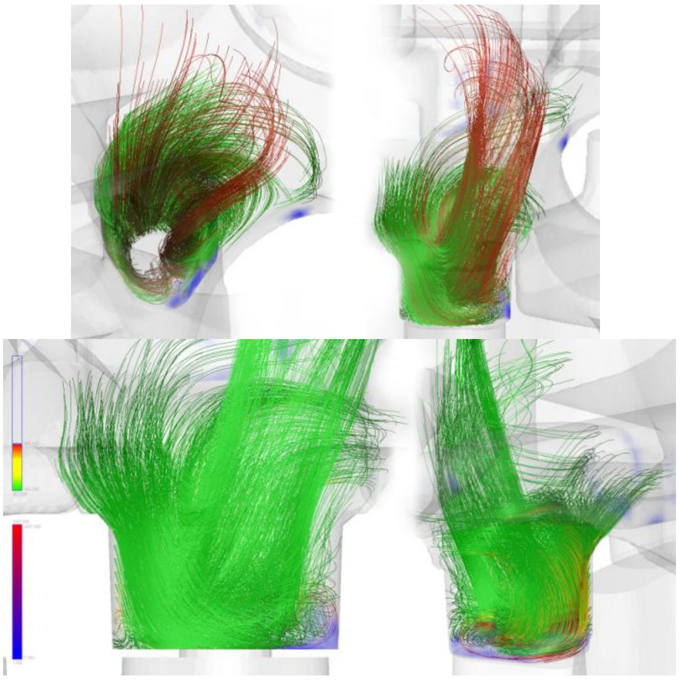


Figure 11: Flow structure at the gasket. *Left*: Outer ring flow structure above the gasket. Only lines with segments leading down are shown. The color represents the step number, so the beginning of a line is drawn green while the ending is red. Blue regions indicate low streamline density. This visualization reveals that the flow comes mostly from one side along the surface down to the gasket and then joins the fast stream from the cylinder block. The bottom of the ring at the other side is hardly reachable for the cooling liquid. *Right*: We have now mapped temperature to the lines color. The sparse region at one side of the ring is overheating because most of the fluid coming down at the other side joins the gasket stream before it can reach the other side of the ring.

This application study of an industrial data set has demonstrated the practical capabilities of our framework. We have applied the flow structure visualization and linking of analysis layers to all examined problems. Compared to the local flow analysis, the exploration and understanding of the data is much easier with the possibility to investigate the flow behavior from a Lagrangian point of view. The semantics necessary for a good formulation of feature definitions and hypotheses about the problem solution is better perceived within a global data context.

Although we are not application specialists, we have tried to provide a detailed analysis in a form that is possible to generalize usefully for other scenarios as well. The cooling jacket is well designed and the problematic regions are only small local spots, not affecting the global efficiency and safety. However, it was not hard to identify and analyze them. In the case of a data set showing more serious problems, it would be an equally difficult task.

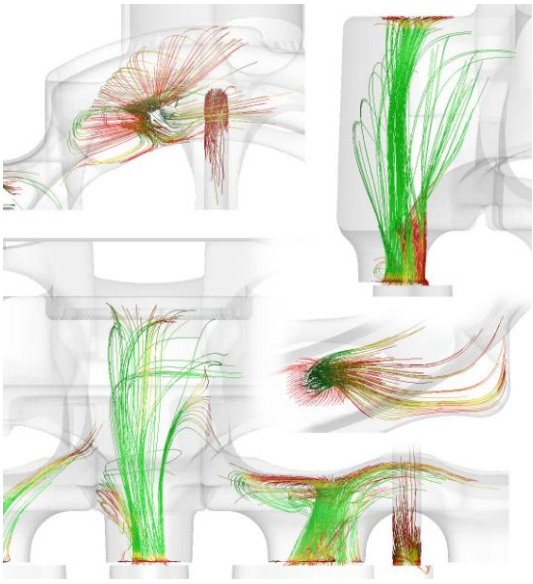


Figure 12: Inner flow at the gasket. Flow structures above various gasket duct viewed from top and side. Red color denotes line segments closer to the hull than 0.5 *cm*. The pitting corrosion is at most probable on the inner surface directly above the wider ducts at cylinders.

7 Conclusions and future work

The contribution of this work is the extension of the SimVis system with an integration based analysis layer that provides the user with means to investigate the long-term flow behavior related to the specified features. We have achieved an improvement of the analysis workflow by enabling the interactive study of relations between flow attributes and structures at once, from various points of view. The concept of linking all layers together has brought increased efficiency to the exploration process. Interactive examination of the flow structures from a Lagrangian point of view has shown to be very helpful for understanding of complex relations within the data. Especially its combination with Eulerian analysis has proved to be very important for apprehension of connections and reciprocity between flow attributes and dynamic structures. The usage of semi-global measures has shown to be very powerful and promising for custom feature specification behind the local scope level. Since the simulation data becomes more and more complex, it is necessary to address such regional descriptors in further research.

There are many challenges for future work following our concept. We plan to extend the set of offered integration methods, interpolation schemes and integral curve types. Computation of integral lines for large unstructured data sets on the GPU is a very promising idea that should considerably speed up the preprocessing stage. However there are still major memory and precision limitations of the graphics hardware to be bypassed. Dealing with more and more complex simulation grids — particularly with moving meshes — will probably require further extensions of the connectivity management. Reduction of memory load by incorporating compression of polylines would greatly help not only in the preprocessing stage. More research on semi-global integral line measures is definitely needed to exploit their possibilities in representation of important flow features.

Acknowledgments

I would like to thank Raphael Fuchs and Philipp Muigg for their long lasting generous support during my work presented in this paper. I appreciate all their comments, advices and patience. Additional thanks go to Helwig Hauser, Andrej Ferko and Matej Novotný. This work has been realized at the VRVis Research Center in Vienna, Austria, funded by the Kplus program. The small cooling jacket CFD is courtesy of Institute for Internal Combustion Engines and Thermodynamics, TU Graz. The large one is courtesy of AVL List GmbH, Graz.

References

1. Raphael Bürger, Philipp Muigg, Helmut Doleisch, and Helwig Hauser. Interactive cross-detector analysis of vortical flow data. In *CMV '07: Proceedings of the 5th International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 98-110, Los Alamitos, CA, USA, July 2007. IEEE Computer Society.
2. Raphael Bürger, Philipp Muigg, Martin Ilčík, Helmut Doleisch, and Helwig Hauser. Integrating Local Feature Detectors in the Interactive Visual Analysis of Flow Simulation Data. In *Proceedings of Eurographics/ IEEE-VGTC Symposium on Visualization 2007*, pages 171-178, 2007.
3. Yuan Chen, Jonathan D. Cohen, and Julian Krolik. Similarity-guided streamline placement with error evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1448-1455, 2007.
4. Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 239-248, Aire-la-Ville, Switzerland, 2003. Eurographics Association.
5. Helmut Doleisch and Helwig Hauser. Smooth brushing for focus+context visualization of simulation data in 3d. In *WSCG '02: Proceedings of 10th International Conference in Central Europe on Computer Graphics*, Visualization, and Computer Vision, volume 10, pages 147-154, 2002.
6. Helmut Doleisch, Helwig Hauser, Martin Gasser, and Robert Kosara. Interactive Focus+Context Analysis of Large, Time-Dependent Flow Simulation Data. *SIMULATION*, 82(12):851-865, 2006.
7. Raphael Fuchs, Bilal Alsallakh, and Meister Eduard Gröller. Delocalized unsteady vortex region detectors. Technical Report TR-2007-030, VRVis Research Centre, 2007.
8. Helwig Hauser. Scientific Visualization: The Visual Extraction of Knowledge from Data, chapter Generalizing Focus+Context Visualization, pages 305-327. *Mathematics and Visualization*. Springer, Berlin Heidelberg, 2006.
9. Helwig Hauser. Visual analysis of differential information. In *ApliMat '05: Proceedings of the 5th International Conference of Applied Mathematics*, 2006.
10. Helwig Hauser, Robert Laramee, and Helmut Doleisch. Topology-based versus feature-based flow analysis - challenges and anapplication. In *Topology-based Methods in Visualization (Proceedings of TopoInVis 2005)*. Springer-Verlag, 2007.
11. Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. In *ACM Trans. Graph.*, volume 24, pages 561-566. New York, NY, USA, 2005. ACM.
12. Robert S. Laramee. *Interactive 3D Flow Visualization Based on Textures and Geometric Primitives*. PhD thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, 2004.

13. Robert S. Laramee, Christoph Garth, Helmut Doleisch, Jürgen Schneider, Helwig Hauser, and Hans Hagen. Visual analysis and exploration of fluidflow in a cooling jacket. In *Proceedings of IEEE Visualization (Vis 2005)*, pages 623-630, October 2005.
14. Liya Li and Han-Wei Shen. Image-based streamline generation and rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):630-640, 2007.
15. Ovidio Mallo, Ronald Peikert, Christian Sigg, and Filip Sadlo. Illuminated lines revisited. In *IEEE Visualization*, pages 19-26. IEEE Computer Society, October 2005.
16. Oliver Mattausch, Thomas Theußl, Helwig Hauser, and Meister Eduard Gröller. Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 213-222, New York, NY, USA, 2003. ACM.
17. Philipp Muigg, Markus Hadwiger, Helmut Doleisch, and Helwig Hauser. Scalable hybrid unstructured and structured grid raycasting. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1592-1599, 2007.
18. Sung Park, Brian C. Budge, Lars Linsen, Bernd Hamann, and Kenneth I. Joy. Dense geometric flow visualization. In *Data Visualization 2005 (Proceedings of the EUROGRAPHICS - IEEE VGTC Symposium on Visualization 2005)*, pages 21-28. Eurographics Associacion, June 2005.
19. Frits H. Post, Benjamin Vrolijk, Helwig Hauser, Robert S. Laramee, and Helmut Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775-792, 2003.
20. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C++ - The Art of Scientific Computing - Second Edition*. Cambridge University Press, 2002.
21. Jonathan C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of the Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pages 61-71, Los Alamitos, CA, USA, 2007. IEEE Computer Society.
22. Tobias Salzbrunn and Gerik Scheuermann. Streamline predicates. *IEEE Transactions on Visualizationand Computer Graphics*, 12(6):1601-1612, 2006.
23. Kuangyu Shi, Holger Theisel, Helwig Hauser, Tino Weinkauf, Kresimir Matkovic, Hans-Christian Hege, and Hans-Peter Seidel. Path line attributes - an information visualization approach to analyzing the dynamic behavior of 3D time-dependent flow fields. In *Proceedings of Topo-In-Vis 2007*, pages 60-74. Springer, March 2007.
24. Thomas Wischgoll and Gerik Scheuermann. Locating closed streamlines in 3D vector fields. In *VISSYM '02: Proceedings of the symposium on Data Visualisation 2002*, pages 227-232. Eurographics Association, 2002.