TIME-CURVATURE AND TIME-TORSION OF VIRTUAL BUBBLES AS FLUID MIXING INDICATORS

Bidur Bohara
Department of Computer Science - Louisiana State University
Baton Rouge, LA 70803

Werner Benger
Center for Computation and Technology - Louisiana State University
Baton Rouge, LA 70803

Marcel Ritter
Department of Computer Science - University of Innsbruck
Technikerstrasse 21a, A-6020 Innsbruck, Austria

Nathan Brener, S.Sitharama Iyengar and Bijaya Karki
Department of Computer Science - Louisiana State University
Baton Rouge, LA 70803

Somnath Roy and Sumanta Acharya
Department of Mechanical Engineering - Louisiana State University
Baton Rouge, LA 70803

ABSTRACT

Massive data sets describing vector fields from computational fluid dynamic simulations on high-performance computers can no longer be visualized directly by displaying the data values at each point in time and space, but require data reduction to analyze the essential properties. We describe early results of work in progress to visualize a dataset of 500GB of raw data, consisting out of multiple time steps from a CFD simulation describing the mixing of fluids within a Stirred tank. The objective is to assess the quality of the mixing of the fluids after some time. We compute pathlines with out-of-core memory management to handle the massive dataset on a single desktop machine and use these to trace the evolution of user-specified initial geometric structures. On top of these structures we compute quantities indicating the mixture, such as curvature and torsion of the pathlines as information complementary to the intrinsic curvature of the evolving time surfaces.

KEYWORDS

Vectorfield visualization, time-dependency, large data, differential geometry, computational fluid dynamics

1. INTRODUCTION

The time-dependent vector field data from computational fluid dynamics can be in the range of a few hundred time steps and be of size in order of terabytes. Visualizing vector fields by extracting key features of flow field is of great importance for most of the scientific field. However, visualizing a huge amount of time dependent data set at once is even more challenging, and in some case doesn’t make sense. The main challenge we have highlighted in this paper is vector field analysis via computation of the curvature measure of flow field along pathlines and time surfaces. Display of pathline curvature on time surfaces provides dynamic information on top of the static surfaces per time step and supports illustrating the flow characteristics in the fluid. Such observations can be helpful for identifying the optimal conditions for mixing of fluids.

Generation and rendering of time surfaces for time-varying vector fields on uniform grids is presented by [Krishnan et al., 2009]. They focus on the generation and rendering of time surfaces with adaptive surface refinement for this simple grid type. Analysis of static vectors field via curvature and torsion is described by [T. Weinkauf, 2002]. They present the idea of computing a curvature field from a 3D vector field, and rendering the curvature measure as color coded on streamlines, via isosurfaces or volume rendering, thereby depicting regions of high curvature in the entire domain.

This paper deals with the more difficult case of data given on a curvilinear multi-block grid, covering a huge amount of data. In contrast to uniform grids that can be kept in memory here more challenging difficulties need to be overcome. The article talks about the theory of curvature measure and its significance on the analysis and understanding of mixing behavior of the fluid system in the Stirred tank system as a means to explore the vast data set. The time-dependent features of fluid flow are computed and rendered as time surfaces and pathlines, the curvature measure of surfaces and pathlines are color coded on the surfaces and lines to provide information about the dynamic evolution of the vector field at each time step.

2. RELATED WORK

In the context of flow visualization, one of the important goals is to visually represent the vector field in clear and meaningful way, presenting the features and characteristics of fluid flow. Lots of techniques already exist for visualizing the flow characteristics. Most commonly used techniques are based on particle tracing and field line based visualization, such as streamlines, streak lines, timelines and pathlines.

In [Robert S. Laramee, 2004], the authors talk about visualizing swirl and tumble flow using three different flow visualization methods, and compared among those methods. Even though, they use 3D vector field they failed to analyze for time-dependent 3D vector field. Another focus of flow visualization is extracting and visually representing the features of flow field. [T. Weinkauf, 2002] talks about visualizing the vector fields using the tangent curves and analyze the properties of curves by evaluating the properties, such as curvature and torsion of the curves. They propose the method of curvature and torsion measures for 3D vector fields, but only at one time slice. [H. Krishnan, 2009] and [B. Bohara, 2010] talks about visualization of time-dependent 3D vector field using time surfaces and streak surfaces. Time surfaces rendered as the natural higher-dimensional extension of time lines is the evolution of a seed surface of particles in the flow of a vector field.

3. FLUID MIXING INDICATORS

When two mutually soluble liquids are mixed together two things happen. First, the liquids are broken up into intermingled clumps. The shapes of these clumps vary depending upon the mixing process. But the average size of the clumps decreases up to a certain range as mixing is continued. This is the macro aspect of mixing which is mainly driven by the mechanical forces. The second aspect is micro-mixing, i.e. molecular inter-diffusion of the fluids across the boundary of their clumps. This process is spontaneous and continues even after the mechanical mixing is stopped and eventually due to this mixture of two soluble liquids becomes completely uniform after a substantial amount of time. However, this molecular mixing is also restricted by the macro mixing process as the break-up of the clump, rate of generation of the surface area of each clump and the topological qualities of the surfaces govern the rate and process of inter-diffusion of the liquids [Ottino, 1989]. In order to develop an understanding of the mechanical macro mixing process, we investigate motion of the fluid particles in the flow domain. The flow domain considered is a mechanically agitated turbulent Stirred tank with down-pumping pitched blade impellers. Water is considered as the working fluid in which a blob of a tracer is injected at a particular location. We look into the pathlines of a number of points associated with the injected tracer. These pathlines show exactly how tracer particles move inside the Stirred tank volume. A good macro-mixing will show the entire volume being filled-up by the pathlines. Now, we look into each individual pathline for understanding the local macro mixing process as well as the potentials of molecular mixing. As discussed earlier, both macro and micro scale mixing are augmented by break-up of the clumps, distortion of the clump-surface area which arises due to large strains on the fluid elements. From the pathline visualizations, we try to qualitatively understand the stretching and distortions on the fluid elements by calculating curvature at different points of the pathline. A higher value of the curvature indicates straining of the fluid element associated with it and also higher residence time of the fluid particle in that particular region resulting into prolonged inter-diffusion of the fluids promoting micro level mixing [Nauman, 2003]. Therefore, the curvatures of the pathlines are of prime interests in order to comment on the potentials of better mixedness at different locations of the tank over the computed time-interval. One other interesting geometric property of the pathline is its torsion which basically measures its deviation from the osculating plane. Torsions show the three-dimensionality of the motion of the fluid particles and indicate on the mixing process away from its plane of motion. Torsion is also a measure of twisting strains on the fluid elements which also promotes the break up rate of the clumps and enhances mixing.

However, a better understanding on scalar mixing can be accomplished by studying the area properties [Ottino, 1989] of the tracer blob volume. Where, the specific rate of generation of the area can give a measure of mixing efficiency. Also different topological properties of the surface area serve as metric of the mixing inside the vessel, such as, curvature and local undulations and depressions of the surface area can be used to identify the roll-up or spread-out of the unmixed fluid sheet. Moreover, curved area can form horseshoe maps resulting into fluid exchange and mixing of one fluid into another. The initial tracer volume breaks up into clumps of smaller volume and disperses through the entire fluid domain. The break-up of the blobs assume an extremely important role in the overall mixing process by diminishing sizes of the unmixed tracers, increasing rate of mixing and providing higher area for molecular mixing. However, the point of break-up of a tracer blob must be a critical point in the flow domain. Hence, a visualization of the flow topology (not the area topology discussed earlier) via velocity vector field can be utilized to identify the regions at which the tracer clumps break down. Hence, the flow feature extraction strategies can be combined with the Lagrangian mixing study to obtain a more vivid description of the mixing process

In differential geometry, a curve *q* is a function that maps a real-valued parameter *s* to a point ***q(s)*** within a differential manifold *M*. Variation of the curve parameter *s* defines the tangential vector $\dot{\boldsymbol{q}}=dq/ds$ along the curve. The second derivative $\ddot{\boldsymbol{q}}=d^2q/ds^2$ determines the radius of curvature of the curve, usually expressed via the curvature κ parameter given by

$$\kappa = {|\ddot{q} \times \dot{q}|}\Big/{|\dot{q}|^3}$$

The third derivative $\dddot{\boldsymbol{q}}=d^3q/ds^3$ is expressed by the torsion parameter

$$\tau = {\dddot{q} \cdot (\ddot{q} \times \dot{q})}\Big/{|\ddot{q} \times \dot{q}|^2}$$

It describes the deviation of the curve from the plane defined by its tangential and acceleration vector. Both curvature and torsion can be computed along streamlines and pathlines of a vector field *v* where the tangential vector is identical to the vector field $\boldsymbol{v}(\boldsymbol{q}(\boldsymbol{s})) = \dot{\boldsymbol{q}}(\boldsymbol{s})$. These scalar measures are thus indicators of the first and second derivative of the vector field itself. Given the first and second order derivatives, i.e. the Jacobian $J^j_i = {\partial v^j}/{\partial x^i}$ and its derivative $J^j_{i,k} = {\partial^2 v^j}/{\partial x^i \partial x^k}$, it is also possible to directly compute the curvature and torsion *field* from a vector field [Weinkauf and Theisel, 2002].

However, since the Jacobian requires three times as much storage space in memory as the original vector field and its derivative nine times as much, it is unrealistic to output these quantities by the simulation code itself, as the vector field itself already occupies 500GB. Computation on the fly on the other hand is hindered by the complexity of the dataset that is constructed by curvilinear coordinates distributed across multiple blocks with non-regular topology at their boundaries. Furthermore, the curvature and torsion field are quantities referring to each time step, corresponding to streamlines, whereas we are interested in quantities describing the time-dependency of the vector field. Thus in our approach we concentrate on computing pathlines from a vector field first, based on user-specified initially defined seeding points, and then computing curvature and torsion quantities along this subset of the vector field as a post-processing step on the pathlines.

4. DESIGN AND IMPLEMENTATION

4.1 Memory Management and Data Model

With datasets as large as 500GB it is mandatory to load data on demand, and to load only those data that are required for a specific numerical operation. Even on machines where 500GB RAM is available, just reading 500GB from disk to memory takes a significant amount of time which impacts turnaround cycles for data analysis and code development. We require data stored in a file format that supports partial file access and to read metadata (describing dataset properties) independent from the numerical data values (describing the actual vector field). These requirements are provided by the Hierarchical Data Format V5 (HDF5), a file format and API developed for HPC applications. Its internal structure is like a file system, but provides additional features such as direct support for n-dimensional floating point arrays, transparently taking care of numerical transformations such as big-endian vs. low-endian and single-precision vs. double-precision (among others). HDF5 is a very flexible and general format, it does not define how a concrete dataset would need to be stored, it only provides the means to do so in one or another way. Choice of the right layout is critical for the final performance; it is essential to layout the data in the file such that not the entire file needs to be read when only partial information might be sufficient. In our version, we use a two-level approach: all data that are related to the same physical time of the simulation are grouped together, and all information that is required to describe the data structure is formulated via meta-data that can be read independently of the data values of the vector field itself. Thus when opening the file only the directory information on how many time steps exist, and at which physical time, is read. Even for a large file of 500GB this required just 4 minutes and 20 seconds when opening the file the first time on our desktop machine (while still under heavy workload of other processes). Subsequent file openings are executed within three seconds as the file's directory structure will be in the cache of the operating system. Actual time slices are only accessed once required by the application, i.e. the integration routine or visualization module, depending on user interaction. Information about the data fields provided per time slice is provided on-demand, including bounding box location of the vector field's blocks. Actual data are only loaded if an integration point falls within the geometry of such a block's bounding box information, leading to an overall nearly interactive experience throughout the entire 500GB dataset on a desktop computer.

In [T. Weinkauf, 2005], the authors present the concept of an out-of-core data handling strategy to process the large time dependent dataset by only loading parts of the data at a time and processing it. Two major strategies presented for out-of-core data handling are Block-wise random access and Slice-wise sequential access. The authors emphasize the Slice-wise sequential access strategy for handling the data given in time slices, however, we have implemented both block-wise access and slice-wise access of time-dependent data while integrating the time surfaces and pathlines.

4.2 Line and Surface Integration Infrastructure

The integration-based visualization is implemented in a visualization environment via plug-ins. The algorithm is split in modules that are responsible for the following tasks:

- Defining seeding geometry or setting initial conditions
- Do integration and refinement
- Rendering

All these modules utilize a strong data library to handle the datasets and geometry efficiently. The integration module is based on a C++ template base class that provides functionality to several kinds of integration modules, such as handling input data, defining the input control layout (and GUI) and the control of the main integration loop. Integration modules, such as streamlines, geodesics, pathlines and integration surfaces can be implemented by specializing template type traits used by the template base class. Different kinds of integration schemes are supported by specialization of different functions of an atomic integration class type trait, for example 4th order Runge Kutta or Dop853 [EH2000].

In the integration module, the input data for one time slice is accessed only once and processed to generate the time surface at that particular time. In case of pathlines, the data is processed to generate the set of points on all the pathlines at that time. We use a sphere as initial seeding geometry represented as sets of points and connectivity information among the seed points. We can also use any other seeding geometry, such as tetrahedron or cylinder. Integration module takes the vector field and the seeding information for every time slice and generates a new time surface represented by set of points and corresponding connectivity information. The connectivity information is used to generate the triangular mesh for surface generation. In case of no surface refinement, the triangular mesh remains constants throughout the time slices. With surface refinement the task of computing a time derivative of the vector-field for new vertices is non-trivial as new vertices don’t have a real predecessor point. We are currently working on this aspect of surface refinement with curvature values.

Curvature is computed by accessing the vector-fields at the previous time slice and the current time slice so is computed as the first order derivative of the vector field over time. Torsion is another important measure of the tangent curves and is computed as the second order derivative of vector field over time. The torsion value for pathlines is computed by accessing vector field for one more time slice than curvature measure. Once the vector field is interpolated at each integration point, the curvature and torsion measures for pathlines are computed by accessing the interpolated vector field along the pathlines.

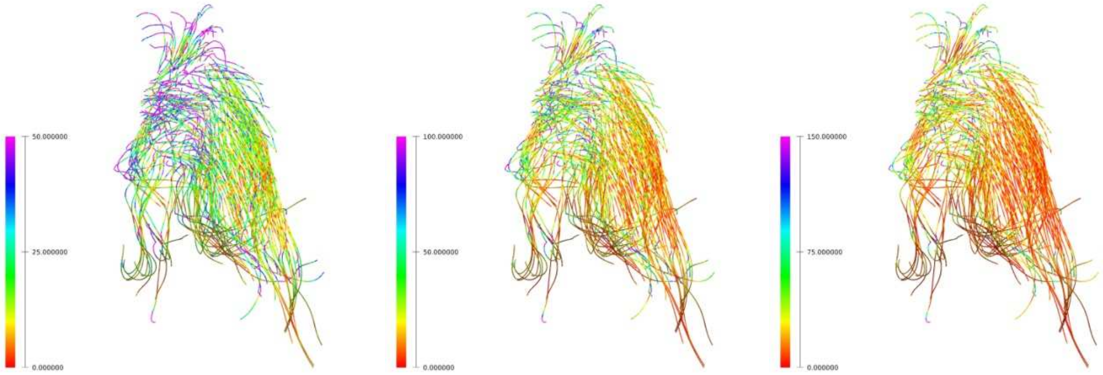

Figure 1. Color coded representation of curvature measure on pathlines for 200 time slices of Stirred tank data. The maximum curvature value represented is 50, 100 and 150 respectively for Left, Middle and Right image. The pathlines are integrated using 4th order Runge Kutta method.

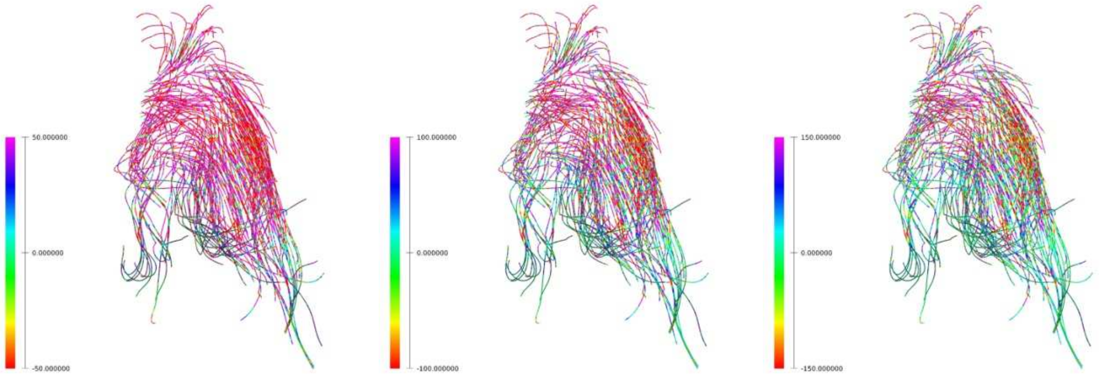

Figure 2. Color coded representation of torsion measure on pathlines for 200 time slices of Stirred tank data. The range of torsion value represented is [-50,+50] , [-100, +100] and [-150,+150] respectively for Left, Middle and Right image.

4.3 Data Interpolation in Curvilinear Multi-Block Grids

The pathline computation algorithm has been made independent on the underlying grid structure. Local array coordinates of the numerical data are used for data interpolation. A helper class does the computation of local coordinates and returns the ID of one block of the multi-block data. Computing local coordinates in a curvilinear multi-block grid is a computationally intense process. The lookup process has been implemented using a coarse and a fine point search. This speeds up the search for the block and the curvilinear grid cell that contains a given point in world coordinates. A KDTree provides candidates of multi-blocks and a uniform grid mapping data structure provides candidates for grid cells to minimize computation overhead.
The KDTree is build once for all the multi-blocks. It stores the center point of each multi-block and provides the maximal bounding box diagonal, which is used as a distance in the range check of the tree. The uniform grid mapping structure is build once for each searched block on demand. It provides a mapping of a uniform grid cell to a small number of curvilinear grid cells that touch or are contained in a uniform grid cell. Once the uniform-grid-mapper has generated this list of curvilinear cells these are candidates to contain the point P at which we want to interpolate the vector field. The next step is to determine which of these curvilinear cells contains the point and then interpolate the vector field at that point. Several alternatives to do this are currently being investigated, including an approach which uses local coordinates and an approach which uses world coordinates, called the "Direct Interpolation". The local coordinate's method employs Taylor approximation and Newton iteration to compute the local coordinates [DS1998], which are then used for interpolation of the vector data.

The Direct Interpolation method uses the world coordinates of the eight vertices of a candidate curvilinear cell to determine if the point is in the cell as follows: It checks each of the six faces of the cell to see if the point is on the same side of the face as a point in the opposite face. If this check is true for all six faces, then the point is in the cell. After the curvilinear cell that contains the point has been found, world coordinates are used to interpolate the vector field at the interpolation point according to the formula

$$\boldsymbol{v}(x,y,z)=\boldsymbol{a}+\boldsymbol{b}x+\boldsymbol{c}y+\boldsymbol{d}z+\boldsymbol{e}xy+\boldsymbol{f}yz+\boldsymbol{g}xz+\boldsymbol{h}xyz$$

where x,y,z are the world coordinates of the interpolation point and $\boldsymbol{v}$ is the vector field. For each of the components of the vector field, the above formula is evaluated at the eight vertices of the curvilinear cell to give eight equations for the eight vector-valued coefficients $\boldsymbol{a}$-$\boldsymbol{h}$. The Gauss-Jordan Method is used to solve these equations for the coefficients and the interpolated value of the vector field component is then calculated from the above formula. A comparison of the performance and accuracy of the local coordinates and Direct Interpolation methods will be future work.

5. RESULTS

The computation for large time-dependent data will be computationally (time and resource) intensive; we only processed the reduced set of the vector field data for the computation of curvature measure of pathlines.

The reduced set of vector field data is obtained after the block-wise and slice-wise access of entire Stirred tank data. Curvature of a line in space indicates a measure of change in direction of a particle moving along the line and as the curvature increases the particles traverses more in space as it moves between two points along its trajectory. The pathlines map the trajectories of individual tracer particles; the curvatures of these pathlines give estimates of interaction of tracer particles with its surrounding fluid. For a pathline with a higher level of curvature at a particular location of the stirred tank, the corresponding particle stays a longer time and interacts with a higher amount of surrounding fluid. This increases the residence time improving mixing at that region of the tank. Figure (1) shows each individual pathline and its corresponding curvature over the tank volume. It can be observed that, after being seeded on the surface of a smaller circular blob, the tracer particles encompass a substantial area of the tank allowing the tracer particles to mix with neighboring fluid. We can see that in a range of 0-100, most of the pathlines show a below 50 curvature value. Whereas, the curvature is as high as 100 at some locations, indicating that better mixing has been obtained as the tracer particles resided there. However, the pathlines show the map of the particle movement both in time and pressure. So, high curvature values (i.e., better local mixing) may occur at different locations and disjoint time instants. In order to realize how the mixing is progressing in time and space, the time surface of a group of particles is drawn for different time instants. These time surfaces show the growth of the injected tracer blob inside the tank at that particular time. Now the time surfaces are contoured with the curvature levels of the pathlines passing through it. Figure (3) shows such a contoured time surface, for two different range of curvature measure. For left image with curvature range 0-100, at a particular instant, the pathlines in the lower portion of the time-surface show higher curvature indicating that particles residing at those locations encompass more space in their vicinity and result into better mixing with surrounding fluid.

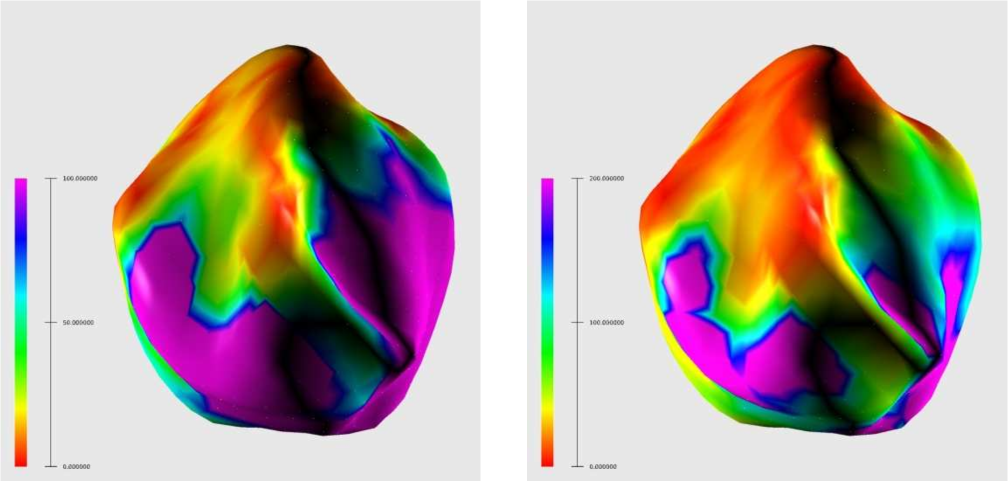

Figure 3. The curvature of pathlines is color coded on the time surfaces. The images show the curvature of points in pathlines at time slice 25, and for curvature measure range of 0-100(left) and 0-200(right)

6. CONCLUSION

Our approach for flow visualization is concentrated in computing pathlines from a vector field, based on user-specified initially defined seeding points, and then on computing curvature and torsion quantities along the subset of the vector field as a post processing step on the pathlines. The computed curve measures are mapped onto the time surfaces generated over the time dependent vector field. The curvatures of the pathlines are of prime interests in order to comment on the potentials of better mixing at different locations of the tank over the computed time-interval. One other interesting geometric property of the pathline is its torsion which basically measures its deviation from the osculating plane. However, the measure of curvature and torsion values of a vector field highly depends upon the seeding position and geometry, as the seed spheres at different places results in different curvature measures for the same time slices. Our approach is not a comprehensive method to find all features and measures of total mixing, but gives the indicative behavior of the fluid mixing. This may result in some missing of key features of flow during the visualization and analysis, as it mostly depends upon the seeding geometry and position.

ACKNOWLEDGEMENT

We thank the VISH development team, among them Georg Ritter, University of Innsbruck, and Hans-Peter Bischof, Rochester Institute of Technology, Amitava Jana and Sanjay Kodiyalam from Southern University, Baton Rouge, for their support. This research employed resources of the Center for Computation & Technology at Louisiana State University, which is supported by funding from the Louisiana legislature's Information Technology Initiative. Portions of this work were supported by NSF/EPSCoR Award No. EPS-0701491 (CyberTools), NSF MRI-0521559 (Viz Tangibles) and IGERT (NSF Grant DGE-0504507).

REFERENCES

E. Hairer, S.P. Norsett and G. Wanner, 2000*, Solving Differential Equations I*, Springer-Verlag Berlin Heidelberg, Germany

D. Stalling, 1998, *Fast Texture Based Algorithms for Vector Field Visualization*, PhD thesis at the Konrad-Zuse-Zentrum für Informationstechnik (ZIB), und freie Universität Berlin, Berlin, Germany

T.Weinkauf and H. Theisel, 2002, Curvature measures of 3d vector fields and their applications. *In Journal of WSCG 2002, International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Plzen, Czech Republic, Number 10, pages 507–514. WSCG, 2002.

Naumann, E. B., 2003, Residence Time Distribution, *Handbook of Industrial Mixing*, Wiley-Intersciences.

Ottino, J. M., 1989, The Kinematics of Mixing: Stretching, Chaos and Transport, Cambridge University Press.

H. Krishnan, C. Garth and Kenneth I. Joy, Time and Streak Surfaces for Flow Visualization in Large Time-Varying Data Sets. *Proc. IEEE Visualization '09, Oct. 2009.*

Bidur Bohara, Farid Harhad, Werner Benger, Nathan Brener, S. Sitharama Iyengar, Bijaya B. Karki, Marcel Ritter, Kexi Liu, Brygg Ullmer, Somnath Roy, Sumanta Acharya, Evolving Time Surfaces in a Virtual Stirred Tank. *In Journal of WSCG 2010. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Plzen, Czech Republic. WSCG 2010.

T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel. Feature flow fields in out-of-core settings. *In H. Hauser, H. Hagen, and H. Theisel, editors, Topology-based Methods in Visualization, Mathematics and Visualization*, pages 51–64. Springer, 2007. Topo-In-Vis 2005, Budmerice, Slovakia, Sept. 29 - 30.

R S. Laramee, D Weiskopf, J Schneider, and H Hauser, *Investigating Swirl and Tumble Flow with a Comparison of Visualization Techniques* in Proceedings of IEEE Visualization (IEEE Vis 2004), pages 51-58, October 15-19, 2004, Austin, Texas

T McLoughlin, R S. Laramee, R Peikert, F H. Post, and M Chen, Over Two Decades of Integration-Based, *Geometric Flow Visualization in EUROGRAPHICS 2009, State of the Art Reports*, pages 73-92, 30 March – 3 April 2009, Munich, Germany