Eyelet Particle Tracing – Steady Visualization of Unsteady Flow

Alexander Wiebel*

Gerik Scheuermann[§]

Image and Signal Processing Group Department of Computer Science University of Leipzig



Figure 1: left: Path lines (blue to red), streak lines (black) and corresponding surface for particles passing through eyelet in synthetic test dataset. right: Eyelet path lines and eyelet path surface for two different positions (marked by arrows) at wing apex in delta wing dataset. Note how the path lines emanating from the eyelet centered above the apex (red arrow) at first lead into the right primary vortex, then into the right secondary vortex and finally turn to the left secondary vortex.

ABSTRACT

It is a challenging task to visualize the behavior of time-dependent 3D vector fields. Most of the time an overview of unsteady fields is provided via animations, but, unfortunately, animations provide only transient impressions of momentary flow. In this paper we present two approaches to visualize time varying fields with fixed geometry. Path lines and streak lines represent such a steady visualization of unsteady vector fields, but because of occlusion and visual clutter it is useless to draw them all over the spatial domain. A selection is needed. We show how bundles of streak lines and path lines, running at different times through one point in space, like through an eyelet, yield an insightful visualization of flow structure ("eyelet lines"). To provide a more intuitive and appealing visualization we also explain how to construct a surface from these lines. As second approach, we use a simple measurement of local changes of a field over time to determine regions with strong changes. We visualize these regions with isosurfaces to give an overview of the activity in the dataset. Finally we use the regions as a guide for placing eyelets.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation; I.6.6 [Simulation and

Modeling]: Simulation Output Analysis; J.2 [Physical Sciences and Engineering]: Engineering

Keywords: 3D Vector Field Visualization, Time-Varying Data Visualization, Flow Visualization, Vector/Tensor Visualization

1 INTRODUCTION

CFD simulations are a widely used tool for prototyping in industry and research. As fluid dynamical mechanisms are of very complex nature, simulations assuming steady flow are insufficient for many problems. With increasing speed and memory size of the computers used to carry out the numerical computations, simulations of unsteady flow have gained more and more importance. Producing time-dependent vector fields covering many time steps, unsteady simulations provided new challenges for visualization. These were replied to with particle tracing, which allowed drawing path lines, streak lines and time lines [10] as well as it enabled the visualization with animated particles [6].

Especially for the development of aircraft prototypes, the simulations' output has reached extreme sizes in recent times. Datasets requiring several terabytes of memory are produced by CFD simulations in this area. However, not only the sheer size of the dataset in the memory poses new problems to be handled by the visualization community, also the very detailedness and mass of data points requires new approaches for visualization. It is well known that direct visualization of such amounts of data may overstrain the reception capabilities of the human eye. In 2D and 3D steady flow this problem is addressed by feature-based and topological visualization, which reduce the number of drawn geometric objects to a

^{*}e-mail: wiebel@informatik.uni-leipzig.de

[§]e-mail: scheuermann@informatik.uni-leipzig.de

minimum while keeping the relevance of these objects at a maximum. Both general approaches have been subject to extensive research that yielded very satisfying results for the cases of 2D steady and unsteady flow and 3D steady flow. However, there is a gap considering 3D unsteady flow. There has been some research that we will review in section 2, but until now there remain many open questions. We did not find any satisfying results on 3D time-dependent topology in literature and the feature-based methods have problems to visualize the 4-dimensional (3D+t) features in only three spatial dimensions.

Displaying 4D information in 3D space is a central issue in visualization of time-dependent flow data. In most cases this problem is handled by animation even though animation only provides transient impressions. It is not possible for the user to get an overview of all time steps while being able to navigate in this view to get a better impression of it. Although many animations allow the user to stop and navigate in the momentary configuration, no context and no steady overview of the data are available to him.

With the work presented in this paper we want to take a first step into the gap of steady visualization of time-dependent flow data. We utilize the well known techniques of path lines and streak lines combined with an intelligent selection and positioning, i.e. we trace a number of particles running through the same point in space for different times. The particles are selected to pass the point at equidistant and equally distributed points in a given time interval. We refer to the entirety of the obtained lines as eyelet lines and call the spatial position common to all these lines an eyelet because the lines look like threads running through an eyelet (see figure 1). The lines and the surface we construct from them represent the complete area that can be reached by particles passing through the eyelet in the considered time interval. We show how the visualization of the area or of single lines lying in it can help understanding the influence of one location on the global flow and the influence of global flow on a single location. The first is achieved by tracing particles forward in time, while tracing them backwards yields the latter.

To give an overview of the activity in the flow we propose to measure the variation of the vectors over time for each data point. Visualizing these variations with isosurfaces we provide the user a steady overview, which he can examine to get insight in the structure of the whole flow over time. As the eyelet particle tracing unfolds its potentials best in regions with high activity, the variation data is used to select interesting positions for eyelets.

2 RELATED WORK

The visualization of unsteady fields is a busy field of research. As mentioned before a central problem for visualization of timevarying data is the additional dimension of time. This is especially true in the case of 3D flow. Most research addresses this problem by animation. The first approaches in this direction were to animate groups of particles [6], for example to show the movement of streak lines as presented by Lane [10]. In order to provide better spatial impressions flow volumes were presented by Becker et al. [2].

More recent publications use dense and texture-based techniques for visualizing time-dependent flow. The first approaches were presented by de Leeuw et al. [3] who extended the use of spot noise for flow visualization to the time-dependent case and by Shen and Kao [16] who extended the well known LIC for unsteady flows (UFLIC). Lagrangian-Euler Advection [8] and IBFV [20] are some of the most recent advances for 2D image-based techniques. These and many similar techniques were also applied to three-dimensional flow [12]. These extensions, however, all have to deal with problems of occlusion and visual clutter.

Another widely used group of methods is topological and feature-based visualization [13]. For unsteady flow, feature-based methods perform a tracking of features over time. Again, in many cases the time is incorporated by animation. Alternatives to directly displaying extracted features are used by Reinders et al. [14] and Garth et al. [5]. Both show the features respectively singularities in a schematic view. While Reinders et al. use their graph view to ease the navigation through single time steps and to show events like birth, death and annihilation of features over time, Garth et al. show the movement of singularities relative to a given axis. Concerning time-dependent vector field topology the most advanced approaches we found in literature were proposed by Tricoche et al. [19] and Theisel et al. [17]. Unfortunately both are only dealing with 2D time-varying fields.

In the context of our aim to visualize 3D unsteady flow using fixed geometry in the spatial domain of the dataset, the only approach we found is the vortex tracking by Bauer et al. [1]. However, in contrast to our direct method, their approach is feature-based.

Another approach showing information from all time steps of a dataset at once are the time histograms presented by Kosara et al. [9]. With this method the change of distribution of values over time is shown by drawing histograms extended to a third dimension (time).

3 LINE-BASED VISUALIZATION OF TIME-DEPENDENT FIELDS

Using lines as primitives for visualization of vector fields is common practice. In the following we will shortly review three types of lines for unsteady vector fields, emphasize their differences and give their mathematical definition to clarify explanations concerning the eyelet path surface given later in this paper (see sec. 5.1).

For the following definitions let $\mathbf{v} : \mathbb{R}^3 \times [t_{min}, t_{max}] \to \mathbb{R}^3$ be a continuous time-dependent vector field. Let $\mathbf{a} \in \mathbb{R}^3$ be the position of a particle in space and let $t \in [t_{min}, t_{max}]$ be a certain time.

Stream Lines Stream lines are integral curves $c_{a,t}(u)$ of vector fields which are tangential to the vectors of a field's domain. They can be interpreted as trajectories of particles in a steady flow. For time-dependent vector fields streamlines are of little use as they stay in a single time step. Thus they do not show actual particle motion but theoretical trajectories of particles with infinite velocity.

$$u \rightarrow \mathbf{c}_{\mathbf{a},t}(u)$$

$$\mathbf{c}_{\mathbf{a},t}(0) = \mathbf{a}$$

$$\frac{\partial \mathbf{c}_{\mathbf{a},t}}{\partial u}(u) = \mathbf{v}(\mathbf{c}_{\mathbf{a},t}(u),t)$$

Here *u* is a time-independent parameter, *t* selects the time from which the vectors **v** are taken and and **a** is the stream-line seed at u = 0.

Path Lines In contrast to stream lines, path lines $\mathbf{p}_{\mathbf{a},s}$ in unsteady flow, indeed, are the paths of moving particles. Path lines are obtained by integration over space *and* time. For steady flow path lines and stream lines are identical.

$$\begin{array}{rcl} t & \rightarrow & \mathbf{p}_{\mathbf{a},s}(t) \\ \mathbf{p}_{\mathbf{a},s}(s) & = & \mathbf{a} \\ \frac{\partial \mathbf{p}_{\mathbf{a},s}}{\partial t}(t) & = & \mathbf{v}(\mathbf{p}_{\mathbf{a},s}(t),t) \end{array}$$

Here s is the seed time.

Streak Lines Streak lines $s_{a,t}$ are imaginary lines connecting the locations of particles that were released into a flow at consecutive time steps. The lines can be observed when looking at

the particles at a certain time *t*. Like path lines, streak lines coincide with stream lines in the steady case.

$$s \rightarrow \mathbf{s}_{\mathbf{a},t}(s) = \mathbf{p}_{\mathbf{a},s}(t)$$
 (1)

Note that t is fixed and s varies.

For our purpose it is worth recalling that stream lines, and thus path lines and streak lines, can not cross each other in steady vector fields. Also recall that this means that only one line runs through each point and each point uniquely identifies the line it lies on. This is why streamlines yield a good overview of non-varying vector fields and are one of the most popular visualization techniques for such data. Many streamlines spread over the whole domain of a steady vector field produce a good impression of the flow's structure. In time-dependent vector fields, however, there is a line running through each position for every point in time, i.e. a single position is contained in an infinite number of lines. Drawing "all" these lines yields a visualization that suffers extremely from visual clutter and occlusion.

4 VARIATION FIELD

In this section we introduce measures for the change of a vector field over time. We subsume the measures under the notion of *variation fields* or *activity fields* and define all these terms mathematically.

Let

$$\mathbf{v}: {\mathbf{x}_0, \ldots, \mathbf{x}_m} \times {t_0, \ldots, t_n} \to \mathbb{R}^3$$

be a discrete 3D time-dependent vector field that is defined for n + 1 particular times $t_0 < ... < t_n$ at m + 1 nodes $\mathbf{x}_0, ..., \mathbf{x}_m$. Let \mathbf{v}_{t_i} be one time step and $\mathbf{v}_{t_i,j}$ the vector at a node \mathbf{x}_j in time step \mathbf{v}_{t_i} . Let a *variation* or *activity* $a_{t_i}(\mathbf{x}_j)$ be a measure for the difference between the values of a node \mathbf{x}_j in two consecutive time steps $(\mathbf{v}_{t_i}, \mathbf{v}_{t_{i+1}})$. Then we define the *variation field* as a field that at every node contains the variations for this node summed over all time steps, i.e.

$$a(\mathbf{x}_j) = \sum_{t_i=t_0}^{t_{n-1}} a_{t_i}(\mathbf{x}_j) \text{ for } j = 1, \dots, m$$

Throughout this paper we will use and discuss the following measures:

Dot Product Variation This variation is computed by accumulating the positive dot products of vectors in consecutive time steps. The norm of the two vectors and the angle between the vectors influences the dot product variation.

$$a_D(\mathbf{x}_j) = \sum_{t_i=t_0}^{t_{n-1}} |\langle \mathbf{v}_{t_i,j}, \mathbf{v}_{t_{i+1},j} \rangle| \quad j = 1, \dots, m$$

This variation correlates to the similarity measures used by Ebling et al. [4] for pattern matching on vector fields.

Vector Variation As second variation we compute the difference vector of the two considered vectors and take its norm. The physical motivation for this variation is that this norm can be thought of as the "force" needed to turn the first vector into the shape of the second.

$$a_V(\mathbf{x}_j) = \sum_{t_i=t_0}^{t_{n-1}} \|\mathbf{v}_{t_i,j} - \mathbf{v}_{t_{i+1},j}\| \quad j = 1, \dots, m$$

We propose to interpret variation fields as "maps" of activity. Using isosurfaces regions of high activity can be separated from those with nearly steady flow. For most flow datasets regions with high activity are of great interest for analysis. Note that for datasets with non-equidistant time steps each variation has to be scaled proportional to the corresponding distance.

5 EYELET LINES

In this section we define the notion of *eyelet lines* for the visualization context. We extend the eyelet lines to *eyelet path surfaces* and give possible applications.

Eyelet lines, in our notion, are a bundle of lines describing flow or vector field properties in general and running through at least one common position (base point, *eyelet*) at different times. This can be formulated mathematically as follows: Let $\ell_{\mathbf{x},\tau}$, defined as

$$egin{aligned} & \ell_{\mathbf{x}, \tau} : \mathbb{R} & o & \mathbb{R}^3 \ & t & o & \ell_{\mathbf{x}, \tau}(t), \end{aligned}$$

be a curve running through a base point $\mathbf{x} \in \mathbb{R}^3$ at time $\tau \in [t_{min}, t_{max}] \subset \mathbb{R}$, where t_{min} is smallest and t_{max} the largest time value in consideration. Then we call the totality of lines $\ell_{\mathbf{x},\tau}$ with $\tau \in [t_{min}, t_{max}]$ eyelet lines $L_{\mathbf{x}}$.

As mentioned in section 3, for path lines and streak lines in timevarying vector fields one point in space can be crossed by different lines. Thus both line types are suitable as basis for eyelet lines and we can specialize the above definition for them as follows. The definitions are straight forward: Eyelet path lines are eyelet lines obtained by path line integration, i.e.

$$\{\ell_{\mathbf{X},\tau}\in L_{\mathbf{X}}|\ell_{\mathbf{X},\tau}=\mathbf{p}_{\mathbf{X},\tau}\}\;,$$

and eyelet streak lines are all streak lines running through a base point, i.e.

$$\{\ell_{\mathbf{X},\tau} \in L_{\mathbf{X}} | \ell_{\mathbf{X},\tau} = \mathbf{s}_{\mathbf{X},\tau} \} .$$

To ease further explanations a particle passing the base point will be referred to as *passing particle* in the following. Also note that eyelet path lines, all throughout this paper, are colored from blue to red with increasing time of the particle passing the eyelet.

5.1 Eyelet Path Surface

Regarding the set of all possible positions of passing particles as a continuum leads to the notion of *eyelet path surfaces*. Every particle running through the base point \mathbf{x} can only move on this surface and, vice versa, every point of the surface is the position of a particle that runs through \mathbf{x} at a certain time.

All points on path lines and streak lines describe positions of passing particles. For path lines these are the positions of one particle over time, for streak lines the points describe the positions of many particles at a single point in time. With the argumentation of the previous paragraph, thus, all path lines and all streak lines running through the base point lie in the eyelet path surface for x. Indeed both, eyelet path lines and eyelet streak lines, build up the whole surface.

Mathematically this can be seen as follows. The eyelet path surface $P_{\mathbf{a}}$ through \mathbf{a} can be defined by path lines as

$$P_{\mathbf{a}}^{path} = \{ \mathbf{x} \in \mathbb{R}^3 | \mathbf{x} \in \mathbf{p}_{\mathbf{a},s}(t), \quad s,t \in [t_{min}, t_{max}] \}$$

and by streak lines as

$$P_{\mathbf{a}}^{streak} = \{ \mathbf{x} \in \mathbb{R}^3 | \mathbf{x} \in \mathbf{s}_{\mathbf{a},t}(s), \quad s,t \in [t_{min}, t_{max}] \}$$

With $\mathbf{s}_{\mathbf{a},t}(s) = \mathbf{p}_{\mathbf{a},s}(t)$ from eq. 1 and taking into account that s and t for $P_{\mathbf{a}}^{streak}$ and $P_{\mathbf{a}}^{path}$ are running through all time values in $[t_{min}, t_{max}]$, it follows

$$P_{\mathbf{a}} = P_{\mathbf{a}}^{path} = P_{\mathbf{a}}^{streak}$$

This is the coincidence of path lines and streak lines in the eyelet path surface. The left image of figure 1 illustrates the coincidence



Figure 2: Inserting a new line in case of diverging behavior and ignoring an existing line for converging case.

of these path lines and streak lines with the eyelet path surface in a synthetic vector field.

As mentioned, the eyelet path surface represents all possible positions of all passing particles. Thus, performing the tracing of particles only in flow direction, the surface covers all parts of space reachable by flow through the base point. Taking only the parts of path lines in account that are located upstream to the base point, the eyelet path surface represents the region where all flow passing the eyelet is originated from. Applications of these interpretations of the surface are discussed in section 6.4. Eyelet path surfaces can also be seen as path surfaces that have a streak line as starting curve. Widening up the eyelet to a line or a small area would lead to a three dimensional counterpart of eyelet path surfaces similar to a flow volume [2].

5.2 Surface Construction

In our implementation the surface is constructed using the eyelet path lines. We move along two neighboring lines and sample them regularly after a constant step size. We connect these points to triangles as shown for lines c and d in the left images of figure 2. Two consecutive points on both lines form a quad that is built up of two triangles. The diagonal edge connects the pair of opposite points that has the smaller distance.

Neighboring lines start at very close points in time (e.g. t_c , t_d) and as the change of vectors along the time axis is continuous the shape of neighboring lines, in many cases, looks very similar. The continuity in time results from the linear interpolation in time that is performed to obtain vectors between the time steps provided by the considered dataset. As differing shape may imply "shift" in arc length and increasing distance between neighboring lines, this is crucial for the quality of the aforementioned way of generating triangles. However, for strongly changing vector fields slow varying shape is not always given. Diverging behavior of the lines is one case where the distance between the lines changes. In this case we insert a new line between the diverging lines. We do this in a way very similar to the approach of Hultquist [7]. Figure 2 illustrates the idea in the left image. The distance between corresponding points on neighboring lines a and c is controlled by a threshold. If the distance exceeds the threshold a new line (b) has to be inserted. Unfortunately, we can not just start a new path line in the middle between the two considered points (like Hultquist does), as this new path lines, due to numerical errors, would not necessarily run through the eyelet point x. Note also that not only the spatial position, but also the position in time would have to be determined for the new start point. We found that this is not a trivial task. So, having not the same option as Hultquist, we start the new path line (b) at the base position **x** for $t_b = 0.5(t_a + t_c)$. We are aware that this procedure means much extra computation time, however, we think it is well spent in order to ensure accuracy.

The new path line is not used for triangulation from the base



Figure 3: Reuse of previously computed line for diverging behavior after convergence.



Figure 4: Split of eyelet path surface originating from eyelet above wing apex due to large angle between line segments.

point on because this would require to restart the triangulation from the beginning, what, obviously would be a tremendous waste of computation time. Instead, we take as many steps on the new path line as we took on a and c until the distance exceeded the threshold. The reached position is then the first of line b used for surface construction. From there on lines a and b, and lines b and c are the neighboring lines considered for triangulation. For the contrary case of converging lines (see figure 2, right) we simply omit the positions on the path line in the middle and regard e and g as neighbors.

Figure 3 shows path lines that at first draw nearer to each other and diverge afterwards. The divergence here is handled as a special case. We do not always compute a new path line but check whether we have previously computed lines in the time interval associated to the two diverging lines. If there is only one such line we use it to compensate the divergence. In the case of more than one previously computed line, like lines 2 and 3 in figure 3, we chose the line with starting time next to the middle of the starting times of line 1 and 2. We do not detect sequences of convergence and divergence, but, in fact, check for already computed lines for every divergence.

Very rapid divergence of neighboring lines is detected and handled as proposed by Hultquist [7]. This means we compute the angle between two considered path line segments and, if large angles show up, stop triangulation between these path lines. We thus split the front of the surface into two separate parts. An example for such a strong divergence can be observed for the eyelet path lines shown in figure 4. The surface is split up into three parts here.

5.3 Eyelet Position Selection

The usefulness of the eyelet lines crucially depends on the position of the eyelet, i.e. on the position shared by all computed path lines. As path lines through a point only change when the flow at and around this point changes over time, the eyelet path lines for steady flow collapse to one line. Intending visualization of time-dependent fields we do not consider steady but unsteady flow here, but also for unsteady flow wide regions of nearly unchanging velocity are possible. The only conclusion that can be drawn from the collapse to one line is the existence of a (nearly) steady field, in general a not very interesting result. Hence the position of the eyelet should be chosen to lie in regions a priori known to be interesting or at least in regions where the field is unsteady.

We found the regions and positions listed in the following to yield insightful eyelet path lines.

- **Edges and Corners of Flow-Passed Objects** Near to edges and corners of objects, flow often splits or changes its behavior. Hence, interesting eyelet position can be found there. The region around the apex of the delta wing (see sec. 6 and figs. 1 and 4) is a good example for this.
- **Vortex Cores** Positions in vortex cores yield eyelet path lines that can help understanding the feeding process for a vortex. The surface constructed from the backwards integrated eyelet lines describes the region where the particles in the core come from.
- **Singularities** Eyelet path surfaces starting near singularities yield insight in their becoming and evolution.
- **Region Behind Flow Passed Object** Turbulence and swirling motion often appear behind objects immersed in a flow. The sources of these behaviors can be studied by positioning eyelets there.
- Active Regions In general, following the argumentation at the beginning of this subsection, regions with high activity serve as interesting eyelet positions. The variation fields presented in section 4 can be employed to find such regions.

Which positions to choose, in general, depends on the task. However, in many cases the positions of some of the mentioned locations coincide. The delta wing dataset discussed later is an extreme example for this, since there the locations of singularities, high activity and the vortex cores of the main vortices coincide.

As eyelet lines are used for studying the influence of one position in space, in most cases one will draw only the lines or the surface for one eyelet. For presentation, however, it may be interesting to show the influence of a number of eyelets. We found that using different color ranges for different eyelets and transparent surfaces, it is possible to create an understandable visualization for around six eyelets. But note that this number strongly depends on the positions of the eyelets and on the considered data because of possible occlusion.

6 **RESULTS, APPLICATIONS AND DISCUSSION**

We tested our methods using real and synthetic datasets. In this section we describe the real datasets and present the results of applying our methods to them.

6.1 Datasets

6.1.1 Delta Wing

We studied a CFD simulation¹ of airflow around a sharp-edged delta wing at subsonic speed (Mach 0.2). The initially already high angle of attack increases over time, resulting in the creation of vortex bubbles, i.e. in the breakdown of the main vortices. The simulation shows the evolution of the primary, secondary and tertiary vortex structures over time and the breakdown of the main vortices





Figure 5: **upper row:** Top view of isosurface of pressure for first and last time step of hurricane lsabel showing its movement throughout the simulation. **lower row:** Isosurfaces in vector variation field and dot product variation field showing the trace of the hurricanes center.



Figure 6: Eyelet path lines integrated backwards from eyelet near land fall of Isabel (North Carolina) and transparent isosurface of dot product variation field.

¹This simulation dataset has already served as subject of study for other authors. See for example [5, 18]

above the wing. For our tests we picked out every tenth of the 1000 time steps. The grid is constant over time, it consists of roughly 3 million vertices and 11.1 million unstructured cells. Altogether, the dataset has a size of 6.3 GB.

6.1.2 Hurricane Isabel

As second realistic example, we tested our methods on data produced by a simulation of hurricane Isabel² from the U.S. National Center for Atmospheric Research. The dataset was provided in context of the IEEE Visualization 2004 contest. It consists of several time-varying scalar and vector variables on a $500 \times 500 \times 100$ rectilinear grid for 48 time steps. Between consecutive time steps there is a time span of one hour. For our tests we used a spatially subsampled version of the data set. The graphical output is scaled in z-direction (vertical) for a more appealing visualization.

6.2 Data Handling and Performance

All computations were carried out on a standard PC with a 64-bit Opteron CPU running at 2 GHz. The virtual main memory of the machine consisted of 3 GB RAM and 22 GB swap space. Having the 64-bit CPU it is possible for us to address more than 4 GB of main memory and thus to load datasets up to 24 GB within a single process.

For computing the variation fields every position of the data has to be processed for each time step. Thus, no strategy for pre-loading time steps is needed. Vector variation and dot product variation both take about six minutes for the whole delta wing dataset. This scales linearly with the size for other data.

It would be desirable to have real-time interactivity for the computation of eyelet lines and eyelet path surfaces. However, at least for large datasets this is not possible due to several reasons. The first reason is that integration on irregular grids using a Runge Kutta scheme is not computationally cheap. Secondly the computation time depends on the number of computed streamlines and thus on the resolution of the surface. Finally, the fact that time-dependent data does not fit into the RAM slows the computation down because the data has to be loaded from disk. Many approaches for visualization of time-varying fields use special data structures for accelerating computations. This is not possible in our case. As eyelet path lines can run through any point in time and space we can not determine any of the needed data in advance to pre-load it. However, as we hold the complete dataset in the virtual main memory the cache of the operating system can yield performance gains when computing path lines that do not differ much. This helps when computing only few lines with small numerical precision for fast tests and afterwards computing a better resolved eyelet path surface. The cells for the first computation will be kept in the cache and accelerate the second integration. As our data structure [11] is organized with arrays and as grid cells of CFD data often are organized in a way that neighboring cells are stored close to each other, not only the cells of previous computations but also their neighboring cells are cached. Our data structure also supports fast point location which is very important for integral curve computation on irregular grids. While the computations for surfaces like in figure 4, where we have over 7 million field lookups for the Runge Kutta scheme, take about two minutes, computing surfaces for nearby eyelets thereafter take less than 20 seconds.

6.3 Variation Field

Four isosurfaces for different isovalues in the vector variation field of the delta wing are shown in figure 7. The first surface around the whole wing reveals no special features. It only shows that the most



Figure 7: Isosurfaces in the vector variation field of the delta wing dataset. Isovalues (a-d): 2(10% of all values), 10, 50(90%) and 200(98%)

activity can be found around the wing, which is clear a priori. In the next two images the centers of the main vortices emerge step by step. For the last image the isosurfaces do not cover the whole vortices anymore, but form around their breakdown bubbles. Hence, the most interesting features for this data set are exactly where the most activity can be found. The volume enclosed by the isosurfaces for large values are thus very interesting locations to place eyelets. As mentioned before, in fact three criteria for interesting locations coincide there: large activity, vortex core line (early time steps) and stagnation point corresponding to vortex breakdown (late times steps). Eyelet path lines started there can be seen in figure 8.

Isosurfaces in the variation fields of the hurricane Isabel data are shown in the lower images of figure 5. The red isosurface indicates a region of high values in the dot product variation field and the blue isosurface encloses high values of the vector variation field. Both surfaces show the path of Isabel's center. As evidence for this interpretation, in the upper two images isosurfaces of the pressure for the first and the last time step are given showing the movement of the center.

Automation We tried to automate the process of isolvalue selection but discovered that this is not possible offhand. Since we had good experience from the delta wing (see fig. 7) we tried to choose the isovalues such that they divided the dataset at 10%, 90% and 98% of the number of values. This, unfortunately, yielded unsatisfying results for some other datasets. Even choosing isovalues in the mentioned way for a single dataset but for different variations not always yielded useful results. However, large isovalues are a good selection for most data sets. The hurricane dataset is a good example for this because the two isosurfaces in figure 5 are at 94% and 98% of the number of values. As automation is not possible, exploration of variation fields has to be interactive and the user has to choose the isovalues. This is no disadvantage, since, as mentioned before, the range of usually useful values is limited and, in addition, fast isosurface extraction can be achieved by using acceleration data structures [15]. Computing the variation field and setting up the acceleration structure can be done automatically in one step in post-processing of the simulation.

²http://www.vets.ucar.edu/vg/isabeldata/

6.4 Eyelet Path Lines

The delta wing appeared to be an interesting example in our studies of eyelet path lines. The vortices dominate this flow field. Thus the first questions that arose were: Which particles enter which vortex and are there positions from where particles passing at different times enter different vortices. In figure 1 eyelet path lines running into different vortices are drawn. While all particles passing the left eyelet run into the left primary vortex, the traces of particles passing the base point central above the apex lead into vortices on both sides. The lines turn from the right primary vortex to the right secondary vortex and finally run into the left secondary vortex. A very similar situation, used to depict the splitting of the surface, is shown in figure 4. However, notice that for the eyelet in figure 4 the lines lead into the vortices in reverse order. The base points of these similar cases are very close to each other but the particles running through them, nevertheless, take very different directions.

In contrast to the vortices of the delta wing, the vortical structure in the hurricane data set moves. Thus, we could not select an eyelet "in" the vortex. However we found an even more interesting selection, namely the location of Isabel's landfall. The lines generated by integrating against the flow direction are shown in figure 6. They represent the paths of particles approaching to the eyelet, i.e. the winds directed to its position. The lengths of the particles with large time values correspond to the velocity of the particles because faster particles cover a larger distance in the same time. Lines for very small start time values (blue lines) are short because the backward integration reaches the first time step, where the integration has to end, very fast. Not considering these, the length of the other lines (violet to red) show the evolution of the hurricane: The wind has high velocities for the approaching hurricane but slows down immediately when its eye reaches the eyelet. Not only the lengths, but also the directions of the lines reveal the advance. At first the wind comes from the ocean, it develops a turn as Isabel approaches nearer and, finally, when the eye of the hurricane reaches the base point of the lines, it attracts air from inland.

In the rest of this section we give possible applications for eyelet lines that aim at particular physical problems.

6.4.1 Origin of particles in Vortex Bubble

Eyelet path lines started inside a vortex bubble, can aid analysis of vortex breakdown. Integrating eyelet lines backwards in time from a position known to lie in a breakdown bubble for some time steps, shows the origin of the material inflating the bubble. For our experiments we started the lines at the position of one of the stagnation points in time step 750 (fig. 8). These stagnation points are known to play a central role for the creation of the recirculation bubble. We computed lines for two different time intervals. For the right image the interval contained all time steps of the dataset. while the lines in the left image correspond to an interval containing only time steps with a pronounced bubble. The left images shows, in agreement with known facts, that the vortex bubbles are fed only from very small regions around the center of the vortices. Blue lines in the right image show how particles running through the base points before the appearance of the bubbles come also from regions of the vortices apart from their centers.

6.4.2 Combustion

The gas and airflow in combustion chambers is often studied to achieve high effectiveness of the reaction process. An appropriate mixture of gas and air is needed for a proper reaction. The worst conditions for the combustion are where only air or only gas is in a region as no reaction will take place there.

Eyelet lines and their corresponding surface can provide a useful tool for analyzing the flow in such regions. Given a region under suspect to contain no air or no gas, one can select points in this region and compute their eyelet lines backwards in time. The lines then provide an overview of the origins of all particles passing the selected point. If the lines only reach the inlet for air respectively only the inlet for gas this shows that, indeed, no air respectively gas will ever reach the point.

7 CONCLUSION AND OUTLOOK

In this paper we have presented two methods for visualizing three dimensional time-dependent vector fields using steady graphical representation, i.e without the use of animation. One method shows the temporal evolution of path lines or streak lines emanating from a certain spatial location called eyelet. We connected these lines to a surface to display the whole area belonging to the flow through the eyelet. The second method proposed in the paper, computes an "activity map" for the data by measuring local changes of vectors. Isosurfaces in the map show regions of high activity and were used to give an overview of the activity in the field. Different measures were discussed. Both methods were combined by taking regions of large variation as first guess for interesting positions of eyelets. Applied to large flow datasets and interesting physical problems our methods proved their usefulness.

As future work it may be interesting to perform the flow localization proposed by Wiebel et al. [21] for each time step and apply the techniques presented throughout this paper to the obtained fields.

For the variation field computation we plan to explore more similarity measures and test their usefulness for describing the global structure of time-dependent vector fields. We also plan to study possibilities for applying our methods to datasets with moving geometry or moving frame of reference.

ACKNOWLEDGMENTS

The authors wish to thank Markus Rütten, from DLR in Göttingen for providing the delta wing dataset. We also wish to thank Mario Hlawitschka for fruitful discussions and precious hints. Further we are thankful to all members of the FAnToM development team for their programming efforts. Finally we thank the anonymous reviewers whose insightful comments greatly helped to improve the paper. Hurricane Isabel data produced by the Weather Research and Forecast model, courtesy of NCAR, and the U.S. NSF. This work was partly supported by DFG grants HA 1491/15-4, HA 1491/15-5 and SCHE 663/3-7.

REFERENCES

- Dirk Bauer and Ronald Peikert. Vortex Tracking in Scale-Space. In Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2002), VisSym'02, pages 233 – 240, 2002.
- [2] Barry G. Becker, David A. Lane, and Nelson L. Max. Unsteady Flow Volumes. In Gregory M. Nielson and Deborah Silver, editors, *Proceedings of the 6th Conference on Visualization* '95, pages 329 – 335, Washington, DC, USA, 1995. IEEE Computer Society.
- [3] Wim C. de Leeuw and Robert van Liere. Spotting Structure in Complex Time Dependent Flow. In Hans Hagen, Gregory M. Nielson, and Frits H. Post, editors, *Dagstuhl '97, Scientific Visualization, Dagstuhl, Germany, 9-13 June 1997*, pages 47 – 53. IEEE Computer Society, 1999.
- [4] Julia Ebling and Gerik Scheuermann. Clifford Convolution and Pattern Matching on Vector Fields. In *Proceedings of IEEE Visualization* '03, Los Alamitos, CA, 2003. IEEE Computer Society.
- [5] Christoph Garth, Xavier Tricoche, and Gerik Scheuermann. Tracking of Vector Field Singularities in Unstructured 3D Time-Dependent Datasets. In Holly Rushmeier, Greg Turk, and Jarke J. van Wijk, editors, *Proceedings of the IEEE Visualization 2004 (VIS'04)*, pages 329 – 336. IEEE Computer Society, October 2004.



Figure 8: Eyelet path lines for two eyelets (arrows) located at stagnation points (time step 750) of left and right vortex breakdown bubbles, and isosurfaces of vector variation field. Starting time values for lines in left image are restricted to time interval with existing bubble, while starting time values for lines in right image are spread over the range of all time steps.

- [6] Andrea J. S. Hin and Frits H. Post. Visualization of Turbulent Flow with Particles. In Dan Bergeron and Gregory M. Nielson, editors, *Proceedings of the 4th conference on Visualization '93*, pages 46 – 52, 1993.
- [7] Jeffrey P. M. Hultquist. Constructing Stream Surfaces in Steady 3D Vector Fields. In Arie E. Kaufman and Gregory M. Nielson, editors, *Proceedings of the 3rd conference on Visualization* '92, pages 171 – 178, Boston, MA, 1992.
- [8] Bruno Jobard, Gordon Erlebacher, and M. Yousuff Hussaini. Lagrangian-Eulerian Advection for Unsteady Flow Visualization. In *Proceedings of the conference on Visualization '01*, pages 53 – 60. IEEE Computer Society, 2001.
- [9] Robert Kosara, Fabian Bendix, and Helwig Hauser. TimeHistograms for Large, Time-Dependent Data. In Oliver Deussen, Charles Hansen, Daniel A. Keim, and Dietmar Saupe, editors, *Data Visualization 2004* - *Eurographics/IEEE TCVG Symposium on Visualization Proceedings*, pages 45 – 54, Konstanz, Germany, May 2004. Eurographics Association.
- [10] David A. Lane. Visualization of Time-Dependent Flow Fields. In Proceedings of the conference on Visualization '93, pages 32 – 38, 1993.
- [11] Max Langbein, Gerik Scheuermann, and Xavier Tricoche. An Efficient Point Location Method for Visualization in Large Unstructured Grids. In *Proceedings of Vision, Modeling, Visualization*, 2003.
- [12] Robert S. Laramee, Helwig Hauser, Helmut Doleisch, Benjamin Vrolijk, Frits H. Post, and Daniel Weiskopf. The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. *Computer Graphics Forum*, 23(2):203 – 221, 2004.
- [13] Frits H. Post, Benjamin Vrolijk, Helwig Hauser, Robert S. Laramee, and Helmut Doleisch. The State of the Art in Flow Visualization: Feature Extraction and Tracking. *Computer Graphics Forum (Blackwell CGF)*, 22(4):775 – 792, 2003.
- [14] Freek Reinders, Frits H. Post, and Hans J.W. Spoelder. Visualization of Time-Dependent Data using Feature Tracking and Event Detection.

The Visual Computer, 17(1):55 - 71, 2001.

- [15] Han-Wei Shen, Charles Hansen, Yarden Livnat, and Chris Johnson. Isosurfacing in Span Space with Utmost Efficiency (issue). In R. Yagel and G. M. Nielson, editors, *IEEE Visualization '96*, pages 287 – 294, San Francisco, CA, 1996.
- [16] Han-Wei Shen and David L. Kao. UFLIC: A Line Integral Convolution Algorithm for Unsteady Flows. In Roni Yagel and Hans Hagen, editors, *Proceedings of the 8th conference on Visualization '97*, pages 317 – 322. IEEE Computer Society Press Los Alamitos, CA, USA, 1997.
- [17] Holger Theisel, Tino Weinkauf, Hans-Chrsitian Hege, and Hans-Peter Seidel. Stream Line and Path Line Oriented Topology for 2D Time-Dependent Vector Fields. In Holly Rushmeier, Greg Turk, and Jarke J. van Wijk, editors, *Proceedings of the IEEE Visualization 2004 (VIS'* 04), pages 321 – 328, October 2004.
- [18] Xavier Tricoche, Christoph Garth, Gordon Kindlmann, Eduard Deines, Gerik Scheuermann, Markus Ruetten, and Charles Hansen. Visualization of Intricate Flow Structures for Vortex Breakdown Analysis. In Holly Rushmeier, Greg Turk, and Jarke J. van Wijk, editors, *Proceedings of the IEEE Visualization 2004 (VIS'04)*, pages 187 – 194. IEEE Computer Society, October 2004.
- [19] Xavier Tricoche, Thomas Wischgoll, Gerik Scheuermann, and Hans Hagen. Topology Tracking for the Visualization of Time-Dependent Two-Dimensional Flows. *Computers & Graphics*, 26(2):249 – 257, 2002.
- [20] Jarke J. van Wijk. Image Based Flow Visualization. ACM Transactions on Graphics (TOG), Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, 21(3):745 – 754, July 2002.
- [21] Alexander Wiebel, Christoph Garth, and Gerik Scheuermann. Localized Flow Analysis of 2D and 3D Vector Fields. In Ken Brodlie, David Duke, and Ken Joy, editors, *Data Visualization 2005: Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization 2005 (Euro-Vis 2005)*, pages 143 – 150. Eurographics Association, June 2005.