Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2004)
O. Deussen, C. Hansen, D.A. Keim, D. Saupe (Editors)

Surface Techniques for Vortex Visualization

Christoph Garth, Xavier Tricoche, Tobias Salzbrunn, Tom Bobach, Gerik Scheuermann

Visualization Group, Department of Computer Science, Technical University of Kaiserslautern

Abstract

This paper presents powerful surface based techniques for the analysis of complex flow fields resulting from CFD simulations. Emphasis is put on the examination of vortical structures. An improved method for stream surface computation that delivers accurate results in regions of intricate flow is presented, along with a novel method to determine boundary surfaces of vortex cores. A number of surface techniques are presented that aid in understanding the flow behavior displayed by these surfaces. Furthermore, a scheme for phenomenological extraction of vortex core lines using stream surfaces is discussed and its accuracy is compared to one of the most established standard techniques.

1. Introduction

The role of stream surfaces in the analysis of CFD datasets has been a minor one from the beginning. Although they are a natural generalization of streamlines, there has been lack of an algorithm that is able to deal with very complicated flow structures. For CFD simulations from applications where these structures dominate, stream surfaces as a visualization tool have been hard to apply and yielded dissatisfactory results. However, the demand for insightful visualization of complex flow data cannot be satisfied by streamlines and similar line-based techniques alone. They suffer from visual clutter if adequate resolution of features is desired. The groundbreaking drawing work of Dallmann [Dal83] has shown that flow structures can be well understood in terms of flow sheets, owing to the fact that they show the behavior of *all* streamlines in the sheet at once. Essentially, stream surfaces can perform the same task and have an enormous potential in visualizing application datasets. In this paper we present several enhancements to the stream surface computation scheme of Hultquist [Hul92] that allow stream surface integration in domains of intricate structures. We provide some examples that demonstrate their usefulness for visualization purposes. Moreover, we investigate a number of useful techniques that extend stream surface visualization beyond mere display.

Among the research topics in visualization, locating vortices in CFD datasets is one of the major challenges, since they are responsible for a large number of interesting flow phenomena. The analysis of datasets can greatly benefit from efficient detection of vortices and evidence of vortex breakdown (i.e. sudden collapse of vortical motion) if this information is accurately visualized. To provide a viable primitive for the visualization of the geometry of vortex cores, we will describe a physically motivated region definition for vortex cores based on the Rankine [Rot00] vortex model. An algorithm is discussed for extracting vortex cores from flow fields starting out from feature lines and constructing a bounding surface for display. Relating vortex cores to stream surfaces, we develop a simple method for the visual verification of assumed vortex cores and the phenomenological determination of vortex core feature lines.

The contributions in this paper are presented as follows:

- We discuss related work in Section 2
- An improved algorithm for streamsurface computation is given in Section 3
- Section 4 details a region definition for vortex cores and an algorithm for its extraction, given a vortex core line.
- Visualization enhancements for streamsurfaces and a method for extracting vortex core lines using streamsurfaces a presented in Section 5
- Some examples of how this work can be applied are shown in Section 6, and Section 7 concludes on the featured work.

2. Related Work

The work in this paper combines flow visualization using stream surfaces with detection and analysis of vortices.

While stream surfaces have not attracted much attention in visualization over the years, vortex detection has been an interesting topic for a long time. We survey both areas of flow visualization in the following.

Stream Surface Techniques Although stream surfaces represent a substantial qualitative step compared to a bundle of stream lines, there exist only few methods for stream surface calculation. The most common algorithm was introduced by Hultquist [Hul92] and is based on an advancing front of discrete streamlines spanning the stream surface. Adaptive front resolution is used to handle converging and diverging behavior of the flow. Streamline integration is based on a second order ODE scheme and a simple insertion and merging heuristic controls the number of front streamlines. The scheme is straightforward to implement, fast, but performs well for simple flows only. In contrast to this local method, van Wijk [van93] uses a global approach. Through advection of a scalar field from a two dimensional manifold (e.g. the domain boundary) through the grid positions by streamlines, the computation of a particular stream surface reduces to isosurface extraction and permits to use fast and reliable techniques, at the price of heavy pre-processing. The starting curves are necessarily limited to isolines of the 2D scalar field. Adopting the advancing front idea of Hultquist, Scheuermann et al. [SBH*01] exploit the existence of an analytic flow solution for linear interpolation over tetrahedral grids. Although automatic adaption to the grid resolution is inherent, the method is computationally intensive and can only be used on tetrahedral grids. Common to all algorithms is a lack of fine-grained control over the generation of graphical primitives. Aside from explicit stream surface integration, methods exist that create the visual impression of a stream surface by using particles, for example [van92]. Because a stream surface is understood as a two-dimensional flow separator, an explicit representation which is not provided by these methods is usually preferred.

Vortex Detection Vortices have played a prominent role in many flow visualization articles and there are several methods designed to locate them. The central problem of every vortex detection and analysis approach is the lack of an exact definition of a vortex. Usually, swirling motion around some central region is used as working definition [Lug96, Rob91], resulting in either a line-based definition or a region-based approach and a corresponding bounding surface.

As far as line definitions are concerned, Banks and Singer [BS95] look for points with low pressure and high absolute vorticity. From there, they walk some distance in the direction of the vorticity vector (predictor) and compute the vorticity at the new position. In the normal plane of the new position, they look for a pressure minimum and take this as a new point of the vortex core (correction). Sujudi and Haimes [SH95] decompose the grid into tetrahedral cells and compute the Jacobian of the linear interpolant in each cell. In case of two complex conjugate eigenvalues, they use the real eigenvector to compute the reduced velocity and find its zero lines. Peikert and Roth [PR00] show that most vortex core detection methods can be reformulated using the concept of parallel vector fields and propose a second order method that detects the locations of zero torsion on a per cell basis, which are then connected to a line feature.

A typical region-based method is the λ_2-criterion by Jeong and Hussein [JH95]. They calculate the Jacobian and decompose it into symmetric part S and antisymmetric tensor Ω. Vortices are then defined as connected regions where the symmetric tensor $S^2 + \Omega^2$ has two negative eigenvalues.

Taking a more geometric approach, Sadarjoen [SP99] essentially describes a algorithm to detect winding streamlines and displays the results using deformed icons. Jiang et al. [JMT02b] apply Sperner's lemma to identify vortices as fixed points of simplical complexes and extend this technique to detect vortical regions. The curvature density center scheme of Pagendarm et al. [PHR99] computes the swirl plane normal in each grid cell and marks the cells containing the deduced center of rotation. It combines both geometric and pattern matching aspects and can probably be considered closest to our approach (sec. 4) among all the ideas mentioned here.

Vortex verification Since all vortex detection methods are subject to delivering false positives, verification is required. Jiang et al. [JMT02a] propose an automatic geometric verification approach that measures the distribution of streamline tangents on a plane orthogonal to the vortex core.

Vortex visualization Although detection algorithms are derived from line and region based vortex definitions, visualization of the latter is unsatisfactory at times. There are a number of widely used techniques such as streamline seeding near the vortex core to display swirling behavior or cutting planes that slice the vortex and show dataset components that ease the recognition. None of these techniques can be attributed to a single author. Moreover, there are approaches, mostly prevalent in fluid mechanics, that visualize physical context. We will not detail them here.

3. An Improved Scheme for Stream Surface Computation

We present an explicit algorithm for integration of stream surfaces that is based upon Hultquist's original idea of advancing a front of connected streamlines through the flow field and adaptively inserting and deleting streamlines where the flow diverges resp. converges [Hul92]. The original algorithm already produces visually pleasing results in some simple cases, but is unable to satisfyingly handle flow of inhomogeneous magnitude. We eliminate this shortcoming by employing streamline integration based on arc length rather than parameter length, which proves to be a more intuitive

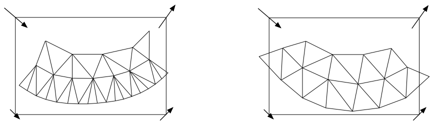
second order ODE arc length parametrization

Figure 1: *improved triangulation resulting from higher order integration combined with arc length parametrization*

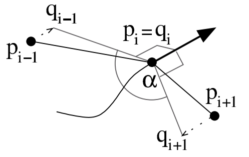

Figure 2: *angular criterion*

and accurate approach for the creation of a graphical representation. Other criteria for refinement are employed to control the density of front streamlines throughout the tracking, taking into account front curvature and the occurrence of singularities in the field. These issues gain importance as stream surfaces are applied for example to the study of vortical structures and flow recirculation.

Flow Inhomogeneity The fundamental task of stream surface integration is to produce a well conditioned triangulation and a good approximation of the real surface. Hultquist achieves this by using a second order ODE solver for streamline integration and selective streamline advancement. This produces triangles whose shape depends on the magnitude of the vector field, even if the resulting surface has no distortions at all requiring higher resolution. We note that graphical precision of a streamline and the numerical step size of the underlying integration scheme are distinct and consider arc length parametrization of streamlines as more suited to graphical representation (resolution is best selected in output domain instead of computational space). Based on an adaptive fifth-order Runge-Kutta integration scheme, we are able to decouple numerically exact streamline integration from the generation of an adequate triangulation of the stream surface (see Figure 1). Thus, surface resolution can be freely parametrized by the user as desired.

Front Curvature A second problem is the rather coarse linear approximation of the surface front, showing up e.g. under strong curvature of the surface perpendicular to the flow. By inserting and deleting streamlines based on an angular criterion applied to the advancing front we keep control over the front resolution. More precisely, strong curvature is indicated by high angular deviation of adjacent front segments, as shown in Figure 2. The basic rule in Hultquist's front propagation scheme proposed the insertion of new streamlines if the width of the quadrilateral at the end of a ribbon between two neighboring streamlines surpasses the height by a factor of two. We add a new rule to enhance this, based on the angle $\alpha = \angle(\vec{q_{i-1}q_i}, \vec{q_iq_{i+1}})$ for points $q_j, j \in \{1..n\}$ being the projections of the front nodes $p_j, j \in \{1..n\}$ into the plane perpendicular to the flow at p_i:

1. Insert a node if the angle exceeds the maximum threshold and the maximum front resolution is not yet reached.
2. Delete a node if the angle falls below the minimum threshold and resolution is still better than the minimum front resolution.

The bounds on the front resolution are supplied by the user.

Algorithm Parameters As mentioned, various resolution parameters can be set to balance the trade-off between number of generated triangles and increased resolution of small features. The size of the triangulation may not seem important at first glance, however, we found the algorithm to easily deal with extremely twisted flow structures like they appear in CFD simulations (an example is the vortex breakdown given in Figure 8). If resolution is too high, the triangulations resulting from those flow fields are too large for interactive display. Hence, a coarser resolution can be imposed.

Parameters influencing the surface quality are front intersegment angle and lower and upper bounds for front node distance (minimum/maximum front resolution). For the latter, values are best chosen on a ratio of 1 to 10 or greater. If the ratio is smaller, the algorithm can oscillate between insertion and deletion of streamlines. A large ratio ensures that converging and diverging streamlines are detected correctly.

The starting curve is given in discretized form as an open or closed line strip and determines the starting discretization of the surface, at best subject to the same resolution constraints that are prescribed for the stream surface. A maximum arc length T_{max} can be set. It is much more intuitive to control stream surface "length" than the maximum parameter of numerical integration applied usually, since it matches the length scale of the dataset.

Surface Parametrization Advancing the front through the vector field has the side effect of producing a parametrization of the surface based on the starting line segment (parametrized as $s \in [0,1]$) and the streamline arc length in flow direction ($t \in [0, T_{max}]$). We will discuss possible applications of this parametrization in section 5.

4. Vortex Core Boundaries

We turn now to an alternative type of surfaces that can be considered the hull of vortices. Therefore we use these surfaces to represent both the shape and the spatial extent of vortices.

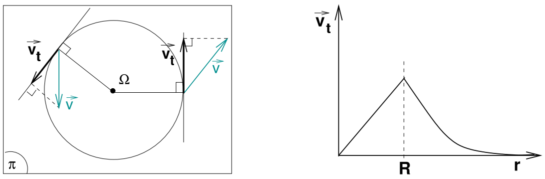

Figure 3: *Tangential velocity (left) and Rankine vortex model (right)*

4.1. Basic Idea

Let us first define tangential velocity v_t, also known as swirl or circumferential velocity. The basic idea is illustrated by the left picture in Fig. 3. Consider the plane Π orthogonal to the vortex axis, the rotation center taken as coordinate origin Ω. The projection of the velocity $\vec{v}$ onto the line perpendicular to the position vector is called *tangential velocity* and is denoted $\vec{v_t}$ in the following. In the same plane we define the *vortex core radius* as the distance from the rotation center of the point where the tangential velocity is maximal for a given radial direction [Lug96]. This definition is best understood when considering the so-called Rankine vortex, see Fig. 3. Observe that rotational symmetry is assumed and tangential velocity is a function of the distance r of a point to the rotation center. Close to the vortex rotation center the vector field behaves like a solid body rotation and tangential velocity grows linearly with respect to r. Beyond a certain distance $r = R$, the tangential velocity decays, inverse proportional to r. This model is very simple. Yet it has the realistic property to have zero tangential velocity for both $r = 0$ and $r = \infty$ and is widely used in practice [Lug96]. It follows from the definition that v_t reaches a maximum for $r = R$ which is defined as the vortex core radius. For our purposes we consider a more general vortex type and do not assume rotation symmetry. Furthermore we only require that a maximum value of the tangential velocity exists in each radial direction. The vortex core boundary is thus defined as the set of points whose distance to the rotation center is equal to the local value of the vortex core radius. Keeping this in mind we now describe our extraction method.

4.2. Algorithm

As mentioned previously the vortex core radius is estimated with respect to a given rotation center and normal plane. Our extraction method takes a polygonal vortex core line as input that we interpret as the loci of rotation centers. This line is assumed given; we describe a streamsurface-based method to this purpose in section 5.1.

Moving along the resampled line we compute around each vertex a polygonal approximation of the actual vortex core boundary. To do so, we need a normal vector that uniquely determines the orthogonal plane Π. We implemented two possible approaches: the normal vector is either provided by the local flow direction or by the tangent line to the feature line. The first one is based on the hypothesis that a vortex core line is a streamline. The second rather interprets the vortex core line as a mean locus of rotation, see section 5. Given the corresponding plane, the vortex core radius must be computed for each discrete angle value $\theta_k = \frac{2k\pi}{N}, k \in 0,..N-1$ around the center. Likewise we restrict tangential velocity computation to sample points along the radial axis. We now process in two steps, as explained next.

Reference Values The first step computes the vortex core radius at few angles $\phi_l = \frac{2l\pi}{M}, l \in 0,..M-1, M << N$. The corresponding positions are depicted with square points in Fig. 4. They will serve as reference values in the second step. Practically, they are obtained by searching the first maximum of the scalar product of the velocity $\vec{v}(r)$ and the coordinate vector e_θ, see Fig. 4. This maximum is computed on-the-fly. Starting from the rotation center we evaluate the sample values and return the first value that is larger than every previous one and the next p values, i.e. the first local maximum over a symmetric p-neighborhood. Sampling is done uniformly. However if we either leave the grid or enter a region where tangential velocity swaps sign, we adaptively reduce the step size in order to accurately sample the field close to the corresponding boundary.

Radius Computation Once we have collected the reference values we need to compute the vortex core radius at the remaining angles. Suppose that we want to evaluate the radius at an angle $\theta_k \in (\phi_l, \phi_{l+1})$, where ϕ_l and ϕ_{l+1} correspond to reference values r_l and r_{l+1}. We restrict the search along the radial axis to a radius interval $[(1-\alpha)R(\theta_k), (1+\alpha)R(\theta_k)]$ where

$$R(\theta_k) = \frac{(\phi_{l+1} - \theta_k)r_l + (\theta_k - \phi_l)r_{l+1}}{\phi_{l+1} - \phi_l},$$

and α is a parameter that bounds possible variations of the vortex core radius from a reference value to the next. The curve corresponding to $R(\theta)$ is shown in Fig. 4 along with the search domain surrounding it. Starting at the lower bound of the interval we look for a maximum in the way described previously. However, since we chose to bound the region considered we can face situations where the actual maxima

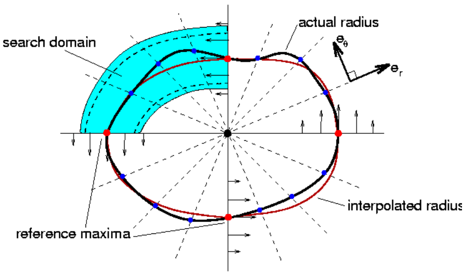

Figure 4: *Boundary curve computation*

lie outside the prescribed interval. This case corresponds to a current maximum lying close to the domain boundary, either interior or exterior. This implies that these current maxima have not been compared with the values of their p-neighbors on one side. We handle this problem by extending the considered interval in the direction of the current maximum, either toward the rotation center or the exterior of the vortex.

Surface Generation The points computed previously form a closed polyline that approximates the vortex core boundary. One curve is generated per vertex of the vortex core polyline. Neighboring curves are then joined with triangles by connecting points that correspond to equal angles on the curve.

Curve Smoothing The resulting triangular surface is in general not smooth, mainly due to interpolation artifacts during radial resampling and to slight errors in the definition of both rotation center and orthogonal plane. To generate a more pleasing visualization we smooth the resulting surface in a post-processing step. To this aim we use the classical umbrella operator [KCVS98] that is very fast and gives satisfying results. Observe that we avoid the natural shrinkage effect by scaling vertices after smoothing to restore the original area enclosed by the curve [DMSB99].

5. Surface Techniques

5.1. Vortex Verification and Core Line Extraction

In the following we introduce a stream surface-based scheme for the visual exploration of vortex structures and the extraction of vortex core lines. This scheme can be applied in situations where automatic schemes (see section 2) deliver unsatisfying or no results and there is a first guess as to the existence and location of a vortex. Moreover, it can be of use in interactive analysis of CFD datasets, due to the relatively low computational cost of stream surface integration as opposed to the high cost of automatic schemes that examine the complete dataset.

The basic idea here is to surround the assumed vortex core with a stream surface and visually observe its rotational behavior to verify the existence of a vortex core. This is achieved by integrating a stream surface from a closed starting curve winding around the vortex axis. The s-parametrization is then color mapped on the resulting stream surface, and rotation of the individual streamlines can be easily confirmed. Given a point p on the assumed vortex core line (or close to it), a good choice for the starting curve (discretized as a polygon P_0) is obtained by taking a circle with small radius (in relation to the dataset dimensions), centered at p and oriented orthogonal to the flow direction $v(p)$ at p (cf. Figure 5).

Once the existence of a vortex is validated, the stream surface can be utilized to compute an approximation to the vortex core line. We note that the lines of constant t-parameter on the surface indicate the integration front location at increasing integration parameter. Since the starting curve was closed and a vortex core was observed, we conclude that the isolines of t on the surface form a closed polygon P_t. To obtain the center of rotation from P_t, a weighted average of the vertices (similar to center of gravity) is taken. The polygon vertices q_t^i, $i = 1, \dots, N_t$ are first sorted in ascending order of the corresponding s-parameters, where N_t need not be fixed due to adaptive triangulation. The average is then computed as

$$C_t \; := \; \frac{1}{2\, N_t\, L_t} \sum_{i=0}^{N_t} q_t^i \cdot \left(||q_t^i - q_t^{i-1}|| + ||q_t^i - q_t^{i+1}|| \right),$$

where L_t is the length of the contour of P_t, and the upper indices are understood modulo N_t. By weighting each vertex with the distance to its direct neighbors and normalizing by L_t, we compensate for varying distance of the polygon vertices due to insertion and deletion of streamlines by the algorithm. Computing C_t for a fixed number of parameters $t_n \in [0, T_{max}]$ and connecting the points with increasing t yields the *gravity line* of the stream surface, which we take as an approximation of the stream surface's center of rotation. In turn, the gravity line closely approximates the vortex core line. Some of the examples from section 6 show how these feature definitions give good results and can be used as input for the vortex core extraction scheme from section 4. The advantage of using a stream surface over a fixed number of streamlines for gravity line calculation is the adaptive resolution of stream surface computation that ensures good approximation even if the vortex core is heavily deformed.

5.2. Plane singularities

To put the idea described in the last section to use, it is necessary to obtain a good starting point p as the center of the closed curve from which stream surface integration is started.

If the flow field of a vortex is projected onto a plane roughly perpendicular to the vortex axis, the planar flow vanishes at the point where the vortex core line intersects the plane. A simple algorithm doing field resampling, projection and a zero search can thus be used to determine approximate

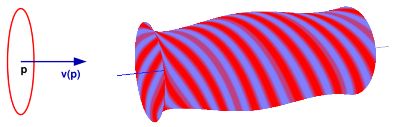

Figure 5: *Left: streamline starting curve P_0 at assumed vortex core line position p, perpendicular to the flow $v(p)$ at p. Right: stream surface colored to perceive rotation around vortex core line*

starting points from very rough guesses. To avoid false positives, the zeros are classified according to their eigenvalues and only those that represent attracting/repelling spirals are taken into account (heuristically, the latter correspond to a cross-section of swirling motion). Refer to Figure 7 (lower right) for an example.

5.3. Stream surface Color Mapping

To further draw on the potential of the generated surfaces, we discuss several methods that enhance the visualization of these surfaces. Corresponding visualization results are shown in section 6.5.

Dataset Resampling Each of the generated surface types is computed in the form of a triangle mesh. Given an appropriate data structure for dataset representation, it is straightforward to extract components of the original dataset at the vertex positions of the generated surfaces. It is standard procedure to slice the dataset by means of a cutting plane and apply a color mapping scheme for certain components of the dataset resampled on that plane. The very same approach can be taken with stream surfaces.

Curvature Given in the form of a triangular mesh, the pure geometry of a stream surface is readily submitted to curvature analysis using discrete curvature operators supplied e.g. by Desbrun [DMSB99]. Strong bending of the stream surface serves as an indicator of inhomogeneous flow behavior, and curvature information aids in comprehending the three-dimensional flow pattern.

Stretching Making use of the parametrization produced by the stream surface algorithm (cf. section 3), we can introduce a quantity called *streamline stretching* that measures the convergence and divergence of streamlines in the surface. Noting that individual streamlines are parametrized by constant $s \in [0,1]$, we find an approximation for the stretching of streamlines S on a per-triangle basis: if s_i, $i = 1,2,3$ are the s-parameters at the vertices of the triangle T, then

$$S := \frac{1}{area(T)} \max\{\, |s_0 - s_1|,\ |s_0 - s_2|,\ |s_1 - s_2|\,\}.$$

In other words, S is an approximation of the length of the integration front in parameter space passing through the triangle T. Again, S can be visualized subject to color mapping.

6. Results and Examples

6.1. Datasets

We tested the presented algorithms on several datasets of aerodynamic CFD simulations performed by the German Aerospace Center (DLR)/Göttingen using their TAU code. All datasets are given as a number of variables (velocity, pressure, density) provided on the vertices of unstructured grids consisting of tetrahedra, pyramids and prisms. Cell-wise linear or trilinear interpolation is assumed in between the vertices.

Stream surface integration as well as vortex core boundary extraction requires a large number of value lookups (cell searching) in the velocity field, thus an efficient data structure is mandatory. We base our point-location algorithms on a *kd*-tree approach recently described by Langbein et al. [LST03]. The datasets are now described in more detail:

ICE train This dataset is the result of a simulation of a high speed train traveling at a velocity of about 250 km/h with wind blowing from the side at an angle of 30 degrees. The wind causes vortices to form on the lee side of the train, creating a drop in pressure that has adverse effects on the train's track holding. The original grid consists of 2.6 million elements.

Delta Wing In order to study vortex breakdown, an unsteady simulation of a delta wing configuration was performed. The simulation was computed without the assumption of symmetry, and totals 1000 time steps that show the formation and breakdown of the primary vortices over time. Since the given methods are applicable to steady flow only, we consider only a single time step that includes vortex breakdown to show the sophistication of the flow structures involved and the aid stream surfaces can provide in understanding them. Grid size: 11.1 million elements.

F6 airplane The simulation contains a steady state flow around a standard plane configuration. Symmetry is made use of for computational efficiency. The grid has 8.4 million cells.

All computations were carried out on a standard PC with 3GB of RAM. The computation times are on the order of few minutes at most.

6.2. Stream Surfaces

Figure 8 gives an impression of the flexibility of the improved stream surface algorithm (section 3). We present an overview of the delta wing dataset with vortex formation at the wing apex and stream surfaces with closed starting curves around the primary vortices (upper left). The starting locations for the latter were obtained using the plane singularity approach (detailed in section 5.2). The lower left image details the breakdown of the primary vortex, sliced by a cutting plane to reveal the inner mechanics. The reversion of flow direction is apparent as the flow folds into the bubble. It is visible that the flow circulates in the bubble many times before leaving it in contorted form. Our algorithm performs this computation well, although a bound on the adaptive resolution has to be imposed as otherwise the resulting triangulation is too large for interactive rendering. The resolution is still more than sufficient to recognize the intricacy of the flow. The lower right image gives a closeup of the apex stream surface. It visualizes the formation of primary, secondary and tertiary vortices above the wing and the flow between them. In the upper right, the ICE dataset has been treated with a stream surface that shows the various vortices created on the lee side of the train.

6.3. Vortex Core Boundaries

Figure 6 (upper row) shows the result of our extraction method applied to the Delta Wing data set. Of particular interest are the different shapes of the primary, secondary and tertiary vortices. They are better understood in the light of Figure 8. As a matter of fact, these three vortical structures coexist on both sides of the wing and are intimately related. Most striking is the squeezed shape of the tertiary vortex. Observe that the extraction scheme is able to distinguish between the different structures although they are very close on the front of the wing. It is worth mentioning that the strong deformations induced by the vortex breakdown entail a loss of consistency of the vortex core boundary. The results of the same technique applied to the ICE train dataset are given in Figure 6 (lower row). The streamlines and the wrapping streamsurface show that the computed tubes confirm the nature and position of the vortex core line as extracted by our surface-based scheme.

6.4. Vortex Core Line Extraction

In order to ensure the utility of our stream surface based vortex core line extraction scheme (cf. sec. 5.1), we employed the algorithm of Sujudi and Haimes [SH95], implemented using the parallel operator of Roth [Rot00] to compare against.

The upper image of Figure 7 gives an overview of the ICE dataset with the vortex core segments extracted using the Sujudi-Haimes analysis. It yields incoherent results and a number of false positives (numerical ghosts of attachment and separation on the train surface, indicated by the arrows). To confirm the existence of the upper vortex, the technique described in section 5.1 has been used and rotating behavior of the streamlines is clearly visible. (Note that the swirling stream surface has been kept short for the purpose of the image.) The vortex core is extracted using the proposed algorithm and results in a much more coherent and smooth feature definition. The lower row of the same figure gives an overview of the delta wing (left), again comparing our method of feature extraction to the results of the Sujudi-Haimes algorithm and showing vortex cores extracted based on these features. Primary, secondary and tertiary vortex cores are extracted reliably and do not overlap. On the right side, plane singularities (cf. section 5.2) above the wing are displayed together with the resulting stream surfaces that show that the singularities do indeed belong to vortices (rotation visible by surface coloring according to s-parameter).

6.5. Stream Surface Color Mapping

Some examples of the techniques described in section 5 are depicted in Figure 9. In the upper left two stream surfaces are extracted from the F6 dataset. Wake vortex formation is clearly visible (compare e.g. to the drawings of Dallmann [Dal83]). The stream surface passing near the wing tip has been colored according to streamline stretching, and one can see that the streamlines do not diverge much until the vortex is forming by a twisting of the surface. The second surface in this image is passing through and split into three fronts by the engine and is color mapped according to resampled pressure values from the dataset. Stretching of the delta wing apex surface is shown in the upper right, and Gaussian curvature for the same surface is given in the lower left. A color mapping of helicity magnitude (lower right) visualizes a strong correlation between increased helicity and the onset of vortex breakdown. It was computed in-place at the stream surface vertices, using resampling of velocity and vorticity and forming the scalar product pointwise for computational efficiency.

7. Conclusion and future work

The objective of the work presented in this paper was to establish surface techniques in the visualization of complex CFD datasets, with a strong emphasis on vortices. Stream surface computation was improved to the point where complicated flows can be treated. We discussed a vortex core region definition that is physically justified and introduced an extraction scheme that can visualize the shape of vortices. Furthermore, we pointed out some applications of stream surfaces going beyond pure visualization, namely visual verification of presumed vortices and phenomenological extraction of vortex core line features. Our results and observations are summarized below, together with some ideas for future work.

Streamsurface computation The described improvements to Hultquist's algorithm make the application of stream surfaces for the visualization of involved flow structures possible. Vortex breakdown serves as an impressive example. In general, selecting a starting location or curve still requires good knowledge of the dataset, or otherwise it is very time-consuming to create good visualizations by trial and error. Turk and Banks have accomplished automatic seeding for streamlines in 2D [TB96], however, for stream surfaces no work on this topic has been published so far. Stream surfaces are also subject to visual clutter if not applied economically. Noting the importance of three-dimensional separation in e.g. aerodynamics, stream surfaces will likely prove a useful tool in creating visualizations of three-dimensional topological structures.

Vortex Cores A scheme for the extraction of vortex core boundaries according to the Rankine model was introduced.

The generated boundary surface serves to represent the shape of the vortex. The method is computationally cheap and produces good results. Some open issues remain, most prominently the influence of the provided feature line on the result. Moreover, the generated surface is not a stream surface. The interaction of both types of surfaces may prove interesting.

Streamsurface Techniques We showed here that a number of insightful visualizations can be provided by stream surfaces aside from mere display. The color mapping techniques show that they can serve as flow adherent cutting-plane like two-dimensional probes for the visualization of flow related properties. Furthermore, it proved valuable to study the geometry of the surfaces more closely. Our phenomenological feature extraction approach performed well in our test cases even though the results given by standard methods were doubtful. It should be emphasized here that our scheme is not meant to replace automatic methods. Rather, we seek to provide a tool that can improve understanding of a dataset through visual exploration and help in creating expressive visualizations. As to future work, the choice of radius employed for the stream surface computation and how it affects the results of the feature extraction scheme should be examined in detail. Again, a connection to the visualization of topological structures should be considered.

Acknowledgements

Foremost, the authors would like to thank Markus Rütten from DLR for close collaboration and many insightful remarks. He kindly provided the datasets considered here. We also wish to thank Ronald Peikert from ETH Zürich for his precious advice on vortex core extraction. Last but not least, our gratitude extends to all members of the FAnToM project at the University of Kaiserslautern for their implementation efforts. This work was partly supported by DFG grants HA 1491/15-4 and HA 1491/15-5.

References

[BS95] BANKS D., SINGER B.: A Predictor-Corrector Technique for Visualizing Unsteady Flow. *IEEE Transactions on Visualization and Computer Graphics 1*, 2 (1995), 151–163.

[Dal83] DALLMANN U.: *Topological Structures of Three-Dimensional Flow Separations*. Tech. Rep. 221-82 A 07, Deutsche Forschungs- und Versuchsanstalt fuer Luft- und Raumfahrt, 1983.

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Computer Graphics Annual Conference Series* (1999), pp. 317 – 324.

[Hul92] HULTQUIST J. P. M.: Constructing Stream Surfaces in Steady 3D Vector Fields. In *IEEE Visualization '92* (Boston, MA, 1992), Kaufman A. E., Nielson G. M., (Eds.), pp. 171 – 178.

[JH95] JEONG J., HUSSAIN F.: On the Identification of a Vortex. *Journal of Fluid Mechanics 285* (1995), 69 – 94.

[JMT02a] JIANG M., MACHIRAJU R., THOMPSON D.: Geometric verification of swirling features in flow fields. In *IEEE Visualization Proceedings* (2002), pp. 307 – 314.

[JMT02b] JIANG M., MACHIRAJU R., THOMPSON D.: A Novel Approach to Vortex Core Detection. In *Data Visualization 2002* (Aire-la-Ville, Sitzerland, 2002), Eurographics Association, pp. 217 – 226.

[KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings Siggraph '98* (New York, 1998), ACM, pp. 105–114.

[LST03] LANGBEIN M., SCHEUERMANN G., TRICOCHE X.: An efficient point location method fpr visualization in large unstructured grids. In *Proceedings of Vision, Modeling, Visualization* (2003).

[Lug96] LUGT H. J.: *Introduction to Vortex Theory*. Vortex Flow Press, Inc., 1996.

[PHR99] PAGENDARM H.-G., HENNE B., RÜTTEN M.: Detecting voritcal phenomena in vector data by medium-scale correlation. In *IEEE Visualization Proceedings* (1999), pp. 409 – 412.

[PR00] PEIKERT R., ROTH M.: The "parallel vectors" operator - a vector field visualization primitive. In *IEEE Visualization Proceedings* (2000), pp. 263 – 270.

[Rob91] ROBINSON S. K.: Coherent motions in the turbulent boundary layer. *Ann. Rev. Fluid Mechanics*, 23 (1991), 601 – 639.

[Rot00] ROTH M.: *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD thesis, ETH Zürich, 2000.

[SBH*01] SCHEUERMANN G., BOBACH T., HAGEN H., MAHROUS K., HAHMAN N., JOY K.: A tetrahedra-based stream surface algorithm. In *IEEE Visualization Proceedings* (2001).

[SH95] SUJUDI D., HAIMES R.: *Identification of Swirling Flow in 3D Vector Fields*. Tech. Rep. AIAA Paper 95–1715, American Institute of Aeronautics and Astronautics, 1995.

[SP99] SADARJOEN I. A., POST F. H.: Geometric methods for vortex extraction. In *Joint Eurographics-IEEE TVCG Symposium on Visualization* (1999), pp. 53 – 62.

[TB96] TURK G., BANKS D.: Image-guided streamline placement. In *Computer Graphics Annual Conference Series* (1996), pp. 453 – 460.

[van92] VANWIJK J. J.: Rendering surface particles. In *IEEE Visualization Proceedings* (1992), pp. 54 – 61.

[van93] VANWIJK J. J.: Implicit stream surfaces. In *IEEE Visualization Proceedings* (1993), pp. 245 – 252.

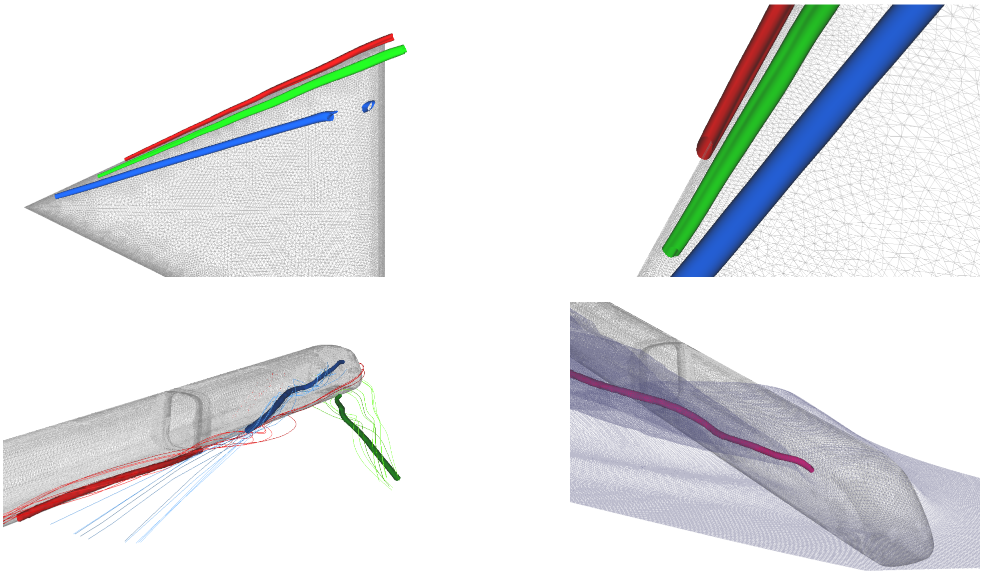

Figure 6: ***Upper row:*** *Vortex cores boundaries above the delta wing. Note the clean distinction between the vortices and the non-trivial shapes.* ***Lower row:*** *Vortex core boundaries in the ICE dataset. Although detected incompletely by the Sujudi-Haimes algorithm, vortex cores could be extracted by our method. The streamlines give a rough impression of how the vortices are created (right). The streamsurface across the nose of the train "wraps" the head vortex (right).*

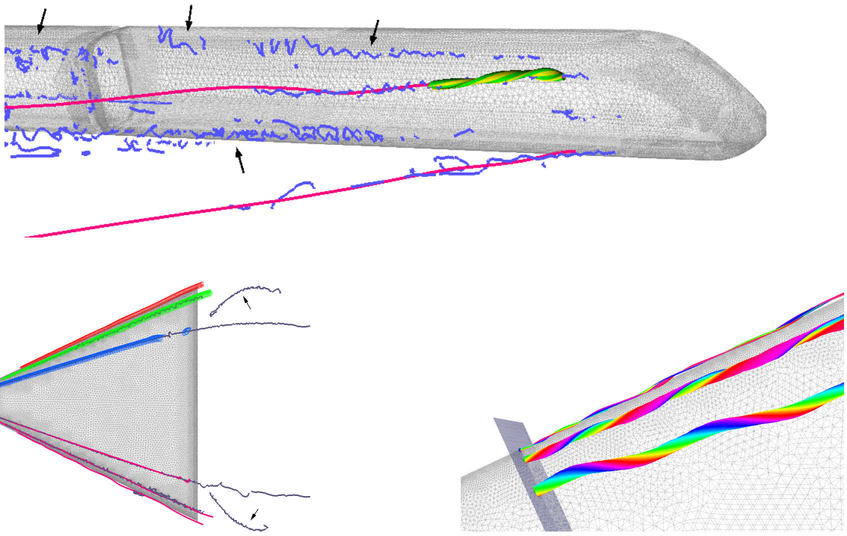

Figure 7: ***Upper image:*** *Results of stream surface based vortex core line extraction on the ICE train: Sujudi-Haimes output with false positives indicated by arrows (blue), vortex core lines computed as gravity lines (magenta), visual verification of the upper vortex using a wrapping stream surface, with color mapping to show the rotation around the vortex axis (yellow/green). Note: not all vortex cores lines have been extracted.* ***Lower left image:*** *Results of vortex core line and vortex core boundary extraction on the delta wing: Sujudi-Haimes output (dark blue, false positives indicated by arrows), vortex core lines from gravity lines (magenta), triangle meshes of vortex core boundaries (primary: blue, secondary: green, tertiary: red).* ***Lower right image:*** *Cutting plane with closed stream surfaces started using plane singularities. Stream surface coloring indicates rotation and confirms the existence of vortices.*

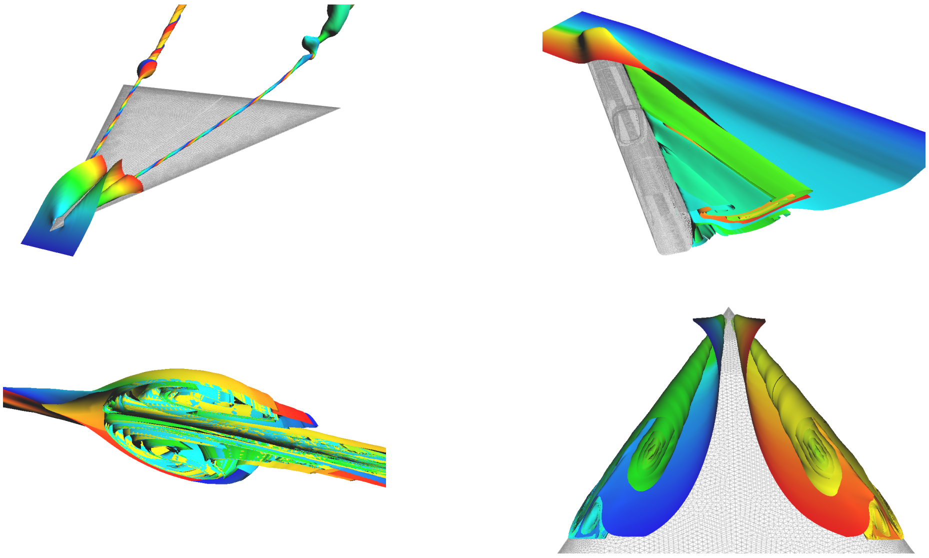

Figure 8: ***Upper left:*** *Overview of the delta wing dataset with vortex creation at apex and the two primary vortices, breaking down differently.* ***Upper right:*** *Stream surface around ICE train showing vortices on the lee side.* ***Lower left:*** *Vortex breakdown cut open, revealing recirculation.* ***Lower right:*** *Formation of primary, secondary and tertiary vortices at wing apex. Note how the shape of the tertiary vortex is strongly elliptic.*

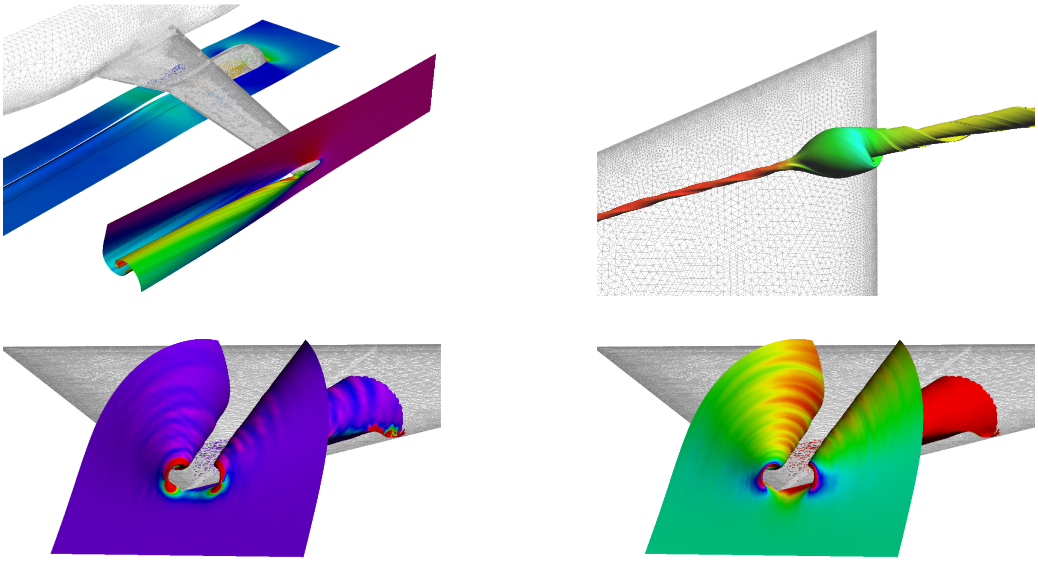

Figure 9: ***Upper left:*** *Stream surfaces in the F6 dataset. Vortex generation at the wing.* ***Upper right:*** *Stream surface showing vortex breakdown, helicity magnitude color mapping. A strong correlation between helicity magnitude and the onset of vortex breakdown is evident.* ***Lower row:*** *Apex surface in the delta wing dataset, colored according to Gaussian curvature (left) and streamline stretching (right), revealing geometric properties of the stream surface.*