A Fast and Robust Method for Visualizing Separation Line Features

Xavier Tricoche[1], Christoph Garth[2], and Gerik Scheuermann[3]

[1] Scientific Computing and Imaging Institute, University of Utah
tricoche@sci.utah.edu
[2] Department of Computer Science, University of Kaiserslautern
garth@rhrk.uni-kl.de
[3] Institute of Computer Science, University of Leipzig
scheuermann@informatik.uni-leipzig.de

The visualization of a three-dimensional viscous flow around an embedded object is typically based on the analysis of its wall shear stress. This vector field defined over the object body exhibits structures that are key to the qualitative evaluation of the surrounding flow. Open separation and attachment lines are of essential interest in aerodynamics due to their adverse effects on the object motion and their implication in vortex genesis. The paper presents a new method for the efficient analysis and visualization of separation and attachment lines on polyhedral surfaces in three-space. It combines local prediction and global feature extraction to yield a scheme that is both efficient and accurate. In particular, it does not suffer from the restrictions induced by assumptions of local linearity and is able to detect features where existing techniques fail. The algorithm is built upon an efficient streamline integration scheme on polyhedral surfaces. The latter is also employed to develop a variation of the LIC scheme. Results are proposed on CFD data sets that demonstrate the ability of the new technique to precisely identify and depict interesting structures in practical applications.

1 Introduction

Modern numerical simulations in Computational Fluid Dynamics (CFD) generate large scale datasets that must undergo qualitative and quantitative evaluation for interpretation. Typically, the analysis relies on the extraction and identification of structures of interest that are used to gain insight into essential properties of the flow for the considered application. In the field of aircraft design in particular, huge amounts of flow data are computed and processed to better understand the properties of design prototypes or to look for optimal configurations, especially during critical flight situations. The usual approach to this analysis is to study the interaction between the three-dimensional air flow around the body and the so-called shear stress vector

field. The latter is tangential to the surface and induces the oil-flow patterns traditionally observed during wind tunnel experiments. Separation and attachment lines are features of key interest in this context. They correspond to one-dimensional loci where the flow leaves or converges toward the body. For aeronautical design this phenomenon is accompanied by adverse effects on lift and drag behavior, in particular during takeoff and landing phases. In automotive engineering flow separation results in a drop in pressure that has negative impact on driving stability. More generally, separation and attachment lines are essential structural features involved in flow partition and vortex genesis. Their automatic extraction and depiction is therefore an important and challenging task for scientific visualization.

In general, the intrinsic limitation of most feature extraction methods is their attempt to extract global structures by means of local analysis. This approach is induced by the need to efficiently address the visualization of very large datasets. Hence, the analysis is aimed at identifying a similarity with some predefined model of the structure of interest. The principal contribution so far to the visualization of separation and attachment lines is the work by Kenwright et al. [5, 6]. Based on considerations inspired by the study of linear vector fields, these authors came up with a simple criterion for local feature identification. Unfortunately, their simple method shows several strong shortcomings, especially in the processing of CFD data sets defined over unstructured grids.

The paper presents a new method for the efficient extraction and visualization of separation and attachment lines in two-dimensional flows defined over arbitrary surfaces in 3D space. The basic idea behind this scheme is to combine local flow probes and global structural information to drive feature search and obtain accurate results fast, even for very large datasets. As a matter of fact, the detection of global features requires the analysis to take global information into account. Since starting a dense set of streamlines over the whole surface to observe and measure their convergence would require a huge computational effort, it is infeasible on typical datasets. However, streamlines are the most natural way to characterize separation and attachment lines. As these lack a formal definition, streamlines are constitutive elements of their empirical characterization. In practice one monitors the flow convergence (resp. divergence) within regions of interest. Concerning our implementation, these regions are characterized by large values of the point-wise divergence operator and are then abstracted to a skeleton of one-dimensional edges by ridge and valley line extraction. The lines obtained can then serve as start positions for streamline integration. To make streamline computation efficient on polygonal surfaces we consider cell-wise constant vector values, resulting in a stable integration scheme that emphasizes attachment and separation behavior.

The paper is organized as follows. Related work is presented in section 2. The technique used for fast streamline integration is introduced in section 3. In particular we discuss integration through so-called singular edges. Additionally, we consider the application of our scheme for fast LIC computation over simplicial surfaces. As mentioned previously, our local feature predictor is the point-wise value of the divergence operator. Its computation is explained in section 4. Section 5 describes the simple though robust algorithm we use to extract ridge and valley lines from

the resulting scalar field. This provides the starting locations required for streamline integration which permits to monitor flow convergence as shown in section 6. Finally, we show some results on two CFD data sets from aerodynamics and comment on the application of our technique.

2 Related Work

In flow visualization, the extraction and visualization of line type features has received much attention in recent years. Besides vortex cores (see [13] for a bibliography), researchers have tried to detect and show separation and attachment lines on bodies immersed in three-dimensional flow. Kenwright [5] made the first major contribution in this area. He proposed a simple and fast method that is suitable for large data sets. A triangular grid and a linear field in each triangle are assumed. The basic idea is that separation and attachment lines can be found in two linear patterns, namely saddle points and proper nodes, where they are aligned with an eigenvector of the Jacobian. Therefore, his method works cell-wise and looks in the corresponding piece-wise linear vector field for the intersection of such lines with the grid cells. The discontinuity of the Jacobian results in disconnected line segments in general. However, inspired by the *Parallel Operator* of Peikert and Roth [10], Kenwright proposed a modified version of his algorithm [6]. It is based on the point-wise evaluation of streamline curvature at the grid vertices. It follows that the extraction of separation and attachment lines reduces to the computation of zero-isolines of the curvature field. As a result one usually obtains connected segments. They must be filtered in a post-processing step to discard false positives [13]. Moreover, dependence on the point-wise computation of the Jacobian introduces a problematic high sensitivity to noise, especially in the case of unstructured data sets. Another approach was used by Okada and Kao [9]. They improve on the classical *Line Integral Convolution technique* (LIC) [2, 1] by first applying a second LIC iteration to sharpen the paths of individual streamlines, and then using histogram equalization to increase contrast. Additionally, they color-code the flow direction which they use to highlight the converging/diverging behavior observed along separation and attachment lines. This method is computationally intensive due to the required LIC processing. Furthermore it does not provide the exact geometry of the feature lines but rather puts emphasis on regions where they are likely to be found. Nevertheless, our method shows in some extent similarities to the ideas used by these authors.

Another aspect directly related to our method is the computation of streamlines constrained to the surface of an object in three-space. This is a classical problem in visualization and many approaches can be found in the literature. Globus et al. [4] mention a simple scheme to keep the streamlines close to the wall along their path. This is done by starting streamlines close to the wall and re-projecting the successive streamline points obtained by integration in 3D onto the object. Max et al. [7] use an Euler method with projection to integrate streamlines on implicit surfaces. Since the surfaces are defined by an implicit function, they can use the function's gradient to define the surface for the projection. Forssell [3] applies LIC to curvilinear

surfaces and gives a corresponding integration scheme. For this, she uses the global parameterization of curvilinear surfaces for her calculation. Battke et al. [1] carry out integration directly on arbitrary surfaces by combining Runge-Kutta with adaptive step-size and linear extrapolation over each triangular cell. Nielson and Jung [8] proposed a computational framework for streamline integration over simplicial grids. Their work is based on the existence of a closed formula for streamlines in linear vector fields. Applied on a cell-wise basis, this permits an exact computation in each cell, reconnected to curves over the whole grid. Unfortunately this technique is quite slow for very large grids. Another mathematical treatment of streamlines on simplicial surfaces is given by Polthier and Schmies [12]. Their method is based on geodesics in accordance with concepts from differential geometry, leading to an adaptation of various numerical integration schemes of varying order for smooth surfaces. We use this technique in the course of our method to obtain smooth and accurate feature lines. Nevertheless, for efficiency reason we adopt an alternative streamline computation scheme to monitor flow convergence as described next.

3 Wall Streamlines over a Simplicial Surface

Before discussing the integration scheme used in our implementation, we briefly introduce the shear stress vector field that is the basic setting in further computations.

3.1 Shear Stress Vector Field

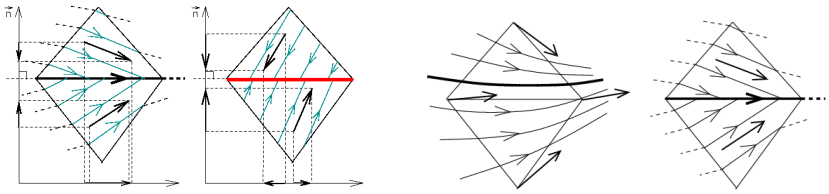
We are concerned with three-dimensional flows that have so-called no-slip boundary condition. This condition is encountered in CFD simulations of viscous flows. It forces the velocity to zero as the body of an embedded object is approached along the surface normal. Therefore, flow analysis around the object deals with the structure of its *shear stress* field [6]. It is a tangential vector field defined over the surface and corresponds to the derivative of the velocity vector field normal to the surface: $\mathbf{v} = J\mathbf{n}$, where J is the 3×3 Jacobian matrix of the velocity field and $\mathbf{n}$ is the local surface normal. Hence it describes how the three-dimensional flow behaves close to the body. Streamlines in the shear stress field are called *wall streamlines*. This definition implies that the integration of wall streamlines takes place in the tangent bundle of the surface. In practice, object boundaries are defined as polygonal surfaces. They are not smooth manifolds since the tangent plane is piecewise constant in each cell but discontinuities occur across edges. Therefore, efficient and reliable techniques to handle numerical streamline integration are needed in this case. In the following we do not assume a particular structure or global parameterization for the surface. Consequently, integration cannot be carried out in a two-dimensional computational space and the results mapped back onto the surface. All things considered the problem to solve is that of streamline integration directly on the surface regardless of its embedding. Furthermore we require computation to be fast enough to be efficiently included in the method presented in section 6. These requirements motivate the scheme discussed next.

3.2 Streamline Integration in Piecewise Constant Vector Fields

Practically, we transform the original shear stress field with vertex-based 3D vector values into a cell-wise constant vector field in which the vectors lie in the tangential plane of the corresponding cells (i.e. resampling to the dual grid and projection into the tangent planes). This choice of procedure comes along with several desirable properties for our purpose. First, we are no longer concerned with the problem induced by the tangent plane indeterminacy at each grid vertex: point-wise 3D vector values lead to different 2D projections onto the tangent planes of incident cells, whereas the tangent plane is well defined for any particular cell. Second, a first-order Euler integration provides the exact solution of the corresponding differential equation. Although a loss of accuracy seems inevitable, the resolution of typical grids provided by CFD simulations allows us to satisfyingly approximate the real path of streamlines in this way. Remark that potential integration instability caused by flow divergence is properly handled in our technique as explained in section 6. The usual requirements on smoothness of a vector field to ensure existence and uniqueness of integral curves are not fulfilled here but separation and attachment patterns extend in this particular setting, as we show next. Moreover, since the path of streamlines is independent of the norm of the underlying vector field (e.g. normalizing a vector field corresponds to a re-parameterization of streamlines by arc length) we normalize the cell-wise vectors for numerical stability concerns in further processing. Now, the integral curve in each cell is a straight line segment connecting two edges. Hence, integrating a curve over the surface corresponds to a cell-wise line clipping, directed by the corresponding vector value. This way of computing streamline was already used in the original LIC technique [2] where the vector information is constant over each rectangular pixel. We consider here a more general problem since we process arbitrary triangles. Nevertheless, provided the connectivity information of the grid, the implementation can be made very efficient.

3.3 The Role of Singular Edges

Since streamlines remain parallel inside each cell, all structural properties of the flow are observed either on the edges or vertices of the grid. We call an edge *singular* if the vector values associated with the cells lying on both sides have opposite normal components with respect to the edge. This means that streamlines reach the edge from both sides but cannot pass through it. If, furthermore, the vectors' components parallel to the edge have opposite directions, no consistent orientation can be decided for further integration and the edge is called *degenerate*. It plays the role of a 1D-singularity since streamlines end there. Refer to Fig. 1(a). If the directions parallel to the edge are consistent, integration can proceed along the edge. This quality of cell-based streamlines is illustrated in comparison to the equivalent situation in the continuous case. The edge corresponds to a contraction (resp. dilation) of the flow (cf. Fig. 1(b)). This property enables streamline integration over piecewise constant vector fields to characterize both separation and attachment in particular cases. As a matter of fact, convergence and divergence are no longer restricted to asymptotic



(a) Singular (left) and degenerate (right) edges (b) Continuous / discrete attachment

Fig. 1. Discrete vector field and singular edges

behavior, but correspond also to the "interception" of streamlines by singular edges, leading in further integration to a one-dimensional flow of all intercepted streamlines from the vertex reached. Of course, the occurrence of this configuration depends on the relative orientation of flow and grid edges. Nevertheless, this interesting attribute plays an important role in the method presented in section 6. Now, once a vertex has been reached along a singular edge, integration must proceed from this position in one of the triangles in its one-neighborhood. Obviously, there is a direction indeterminacy. We first exclude both triangles sharing the considered singular edge. Now, each of the remaining triangles incident to the vertex is a potential candidate to proceed if its vector value lies within the angular domain bounded by both of its edges incident to the vertex, see left configuration in Fig. 2. Both triangles marked *W* are

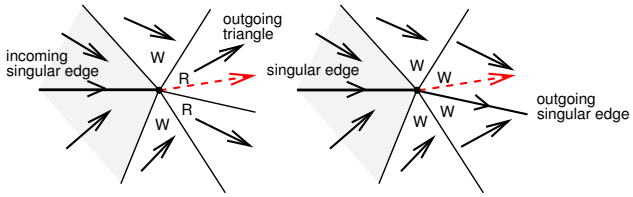


Fig. 2. Streamline integration through a vertex

discarded because their vector values lie outside the angular domain as opposed to the triangles marked "R" that allow for further integration. Practically, we solve the remaining indeterminacy problem by always taking in first place the sector pointed by the vector value originally assigned to the vertex (dashed arrow in Fig. 2). If this sector does not satisfy the angle criterion, we select one of the satisfying neighbors closest to this direction. If no such neighbor exists, we have either reached a singular vertex (in which case integration is terminated) or integration has to proceed along an additional singular edge which is, as well, selected closest to current direction. Refer to right configuration in Fig. 2.

3.4 An Alternative Approach to Line Integral Convolution on Simplicial Surfaces

The streamline integrator on simplicial surfaces derived above is straightforward and computationally cheap. It can thus serve as a building block for the adaption of streamline-based visualization schemes that are usually applied to two-dimensional planar fields. As an example, we consider Line Integral Convolution. Since this technique requires the computation of a great number of streamlines, a fast scheme for streamline integration is mandatory.

In the following, we describe an application of the basic LIC idea [2] to the wall shear stress vector fields, with some modifications. In the common variants of LIC algorithms that deal with non-planar surface grids, a texture is mapped onto the grid to achieve a resolution high enough to recreate the visual impression of oil droplet smearing. In most cases, the grid resolution is not sufficient to convey this effect. However, texture mapping on surface grids is only simple in the case of curvilinear grids. There have been approaches that divide arbitrary surfaces into rectangular parameter domains, e.g. [14], but they are tedious and do not work in all cases. Furthermore, computational effort is large.

We propose a simple yet effective adaption of the original LIC algorithm in the form of three modifications:

- We abandon the idea of mapping a texture on the grid and instead employ a cell-based scalar field over the original grid that carries the scalar values used for convolution.
- Convolution is carried out directly on this field using our streamline integration scheme. Since the visited cells are naturally obtained from the algorithm, it is straightforward to implement the convolution. Furthermore, arbitrary grids are tractable.
- By subsequent subdivision of the cells the grid is refined until the visual representation of the triangles is small enough to give a sufficient resolution. By imposing an upper bound on the area of triangles, the subdivision is adaptive and results in nearly uniform resolution over the whole surface. This constraint can be chosen in such a way that individual triangles encompass only a small number of pixels (ideally one pixel), so that the rendering of the refined scalar field over the surface results in a LIC-like image.

As usual, the initial scalar field is seeded with white noise. The common improvements (e.g. the Festals technique [15]) are easily applied in this context. Although it might seem a disadvantage to create surface triangulations of large size, in practice the size of individual triangles is limited by the fact that the maximum resolution required is somewhat lower than the screen resolution of a typical monitor. Thus the number of triangles is on the order of 10^6, a number well within reach of any algorithm and modern graphics hardware. Remark that streamline integration is not slowed down much by the increased grid size, since no cell location is performed and connectivity information is used instead.

4 Local Predictor

As mentioned previously, the basic idea behind our method is to build feature line extraction on top of a convergence monitoring of the flow. Now, for such an approach to be feasible at all, we need a way to restrict computation to regions of interest, i.e. those regions that lie close to the separation and attachment lines contained in the data set. Hence we need a reliable local predictor that indicates where streamlines might show converging behavior and decreases complexity by several orders of magnitude. More precisely, by predictor we imply a scalar field defined over the whole domain that indicates (either for separation or attachment) which regions are most likely to lie close to line features.

In fact, implicit in the definition of separation or attachment behavior is the notion of contraction and dilation of the flow that occurs normal to the flow direction. This effect is responsible for the convergence of neighboring streamlines. A standard operator to measure flow contraction (resp. dilation) is the divergence. For continuous vector fields, it is defined at each position P as the amount of flow generated in an infinitesimal region around P. In a Cartesian basis of the plane, it is given by the expression

$$(\text{div } \mathbf{v})\,(P) = \frac{\partial}{\partial x} v_x + \frac{\partial}{\partial y} v_y$$

To express divergence in the neighborhood of each vertex of a simplicial surface S, the local geometry around the point must be taken into account. In our implementation we use the formula proposed by Polthier et al. [11] that is expressed as follows for a given position p_i:

$$(\text{div}_S\ \mathbf{v})\,(p_i) = \frac{1}{2} \sum_{e_j \in \partial *(p_i)} \int_{e_j} < \mathbf{v}, \mathbf{n}_j > ds,$$

where $\partial *(p_i)$ is the oriented set of edges e_j opposite to p_i in its incident triangles, and $\mathbf{n}_j$ is the outward pointing normal of edge e_j. Since vector values are provided cell-wise, the computation is straightforward.

5 Ridge and Valley Lines Extraction

Divergence computation results in a scalar distribution over the grid. We saw previously that these values are related to the converging (resp. diverging) behavior of the flow. Now we need to deduce from this scalar field which regions are most interesting for further processing. Since we want to lower the complexity of our convergence monitoring, we choose to extract the so-called ridge and valley lines and to focus on them in the following.

For a scalar field interpreted as a height field, a ridge or valley line is defined as the set of points where the slope is locally minimal compared to points of the same elevation. Ridge and valley line extraction is a classical task in image processing. They are interpreted as edges in a scalar picture. The existing method in that context

are however of little help for our problem since they typically assume structured grids and require first and second order derivative computation [13]. This cannot be achieved numerically in a satisfactory way on a scalar field obtained itself by local estimation of a derivative. For this reason, we adopted an alternative, much easier approach that is fast and gives satisfying results.

We reformulate the definition of ridge and valley lines as follows. Ridge (resp. valley) lines are curves through the domain of definition of a scalar field. They start at local maxima (resp. minima) and minimize descent (resp. ascent) along the curve. In our discrete setting over triangular grids, the corresponding algorithm starts at vertices corresponding to local maxima (resp. minima) and proceeds the ridge (resp. valley) line extraction towards the direct neighbor with maximum (resp. minimum) value. The resulting line connects vertices via the edge segments of the given triangulation.

Since the data at hand is typically noisy (see above), some improvements are necessary to use this method in practice.

- First, we want to restrict ridge line extraction to major features and discard minor ones, hence we must not take into account local extrema due to high frequency oscillations. In practice, we restrict the starting points of line extraction to vertices that are extrema within a large neighborhood surrounding them.
- Second, we want the extracted feature lines to be as straight as possible and to avoid u-turns and self-intersections. Therefore we impose an angle criterion for the acceptance of new segments, given by the mean direction followed during the last few steps. Moreover, we exclude from further processing every vertex that is a direct neighbor of a vertex selected previously.

Modified in this way, the algorithm is very fast, straightforward to implement and robust to noise.

6 Accumulation Monitoring

Once regions of interest have been determined, streamlines are started there. Our scheme then monitors 1D flow convergence resp. divergence through streamlet integration. What is meant here corresponds to the usual empirical characterization of separation and attachment lines as asymptotic limits of streamline accumulation represented by the paths of limit streamlines. Our method consists of two successive steps, as described next.

Cell-wise Accumulation

Practically, we assume that the ridge and valley lines of the divergence provide a coarse approximation of the actual feature lines. Hence, a correction that provides the actual line position is needed. Motivated by the previous remark we choose to measure the flow convergence from the ridge and valley lines on a cell-wise basis.

Observe that ridge lines (positive values of the divergence) are associated with backward convergence (attachment) whereas valley lines (negative values) are related to forward convergence (separation). Practically a cell-wise scalar field accounts in each cell for the number of streamlines that were integrated through it. Each hit corresponds to a '+1' or '-1' value, depending on the sign of the divergence at the starting position. Since we expect the starting locations to lie close to the search region, we only integrate streamlines along a short arc length to highlight the converging behavior. The cell-wise scalar field obtained is converted to a point-wise field using a basic mean value computation.

Feature Line Extraction

A simple idea to obtain the feature lines from the resulting accumulation scalar field would be to apply again the algorithm for ridge and valley line extraction previously discussed in section 5. However, this choice of procedure has two shortcomings. The first one is that the lines obtained in that way provide no guarantee to follow the flow direction as required by our definition. The second one is a direct consequence of our definition of ridge and valley lines: they are constrained to follow the edges of the triangulation which is a coarse approximation of a streamline.

Flow-driven Ridge and Valley Lines

We solve the first problem with a slight modification of the scheme of section 5. We obtain a flow-driven ridge/valley line extraction by processing as follows. Starting at local maxima of the point-wise accumulation field we iteratively move along the grid edges by taking both the values of the 1-neighbors (like before) and the local flow direction into account. The principle is explained in Fig. 3. To obtain a balance

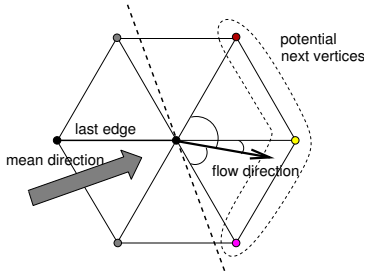


Fig. 3. Flow-driven ridge/valley line extraction

of both scalar value and flow parallelity we perform a simple angular weighting by multiplying the scalar values defined at the vertices of the 1-neighborhood by the cosine of the angles between corresponding edges and local flow direction. Remark that the mean direction of the last few steps is still used to discard vertices inducing a u-turn as shown by the dashed line.

Final Extraction

Given the polyline description of the flow-driven ridge and valley lines, we obtain the actual path of separation and attachment lines by integrating streamlines starting at the upstream (separation) resp. downstream (attachment) end of the ridge and valley lines, and directed toward the converging direction. Integration is terminated when the streamline reaches the vicinity of the other end. This is valid since, according to previous processing, ridge and valley lines are expected to lie at most one cell away from the actual feature line position. Remark that numerical integration in this case is carried out using the geodesic-based technique of Polthier and Schmies [12]. In that way final results are both smooth and very accurate, due to the underlying fourth-order Runge-Kutta scheme with adaptive step size control.

7 Results

To demonstrate the ability of our method to properly extract separation and attachment lines for practical applications we consider in the following two CFD data sets. In both cases we propose a comparison of our results with those obtained using Kenwright's method.

7.1 Delta Wing

The first data set is a steady simulation of airflow around a delta wing at 25 degrees angle of attack. The grid consists of 1.9 million unstructured points forming 6.3 million unstructured elements, 3.9 million tetrahedra and 2.4 million prisms. The delta wing itself is made up of about 81k triangles. We focus on the shear stress vector field defined over the wing that we compute according to the formula mentioned in section 3. Applying Kenwright's method, we get the results shown in Fig. 8, left picture. Observe that a strong pre-smoothing step was necessary to permit a satisfying Jacobian computation. However the results have poor quality due to disconnected segments, shifted features and numerous false positives. The successive steps of our method are shown in Fig. 4. The upper left picture illustrates ridge and valley line extraction from the divergence scalar field. It can be seen that the local divergence computation leads to noisy values. This induces zigzag paths for the lines obtained. However their global aspect provides a satisfying approximation of the features' position as shown in the upper right picture: streamline integration is carried out (either forward or backward) starting along ridge and valley lines until convergence is reached. The resulting cell-wise scalar field accounts for streamline accumulation and is next submitted to flow-driven ridge line extraction, see lower left picture. Smooth streamline integration along the corresponding ridge and valley lines finally gives the exact position of separation and attachment lines. These results are shown again in Fig. 6 together with a LIC texture of the shear stress computed with the technique presented in section 3.4.

7.2 High Speed Train

The second data set corresponds to a single time step of an unsteady simulation of the German train ICE. In this case, the train travels 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. We restrict our considerations to the front wagon that contains 53k triangles. For comparison, the results of Kenwright's method are shown in Fig. 8, right picture. Previous remarks related to pre-smoothing apply here too. In this case, the results are even worse than for the delta wing. We obtain a lot of disconnected segments and most of them would have been filtered out by a simple check on flow parallelity. Moreover the feature line lying on the nose of the train is significantly shifted toward the middle. Again, the successive steps of our method can be seen in Fig. 5. The main difference with previous data set is the presence of weak features on both sides of the wagon that correspond to slow flow divergence. Practically we obtain for these features ridge and valley lines of the divergence that lie fairly far away from their actual position. This implies that the correction step associated with flow monitoring must follow streamlines along a longer path to detect the converging behavior. Another consequence is the fuzzy resulting accumulation scalar field around the features. Nevertheless, our flow driven ridge line extraction is able to properly track their path as illustrated in the upper and lower right pictures. Final results are shown along with a LIC texture in Fig. 7.

8 Conclusion

We have presented a new method for efficient and robust extraction of separation and attachment lines on arbitrary simplicial surfaces embedded in three-dimensional space. Our approach combines a local predictor and a global correction step. We use fast streamline integration to efficiently monitor flow convergence resp. divergence. The original point-wise vector field is transformed into a cell-wise one which both speeds up computation and permits to characterize separation and attachment behavior locally. We show how to use this technique to produce fast LIC textures on arbitrary surfaces. Applied to two realistic CFD data sets, our new method proved able to precisely detect interesting features, even in cases where flow convergence occurs weakly, on large scales. Such structures are missed by existing techniques, as shown by our comparison with Kenwright's standard scheme, which has difficulties on unstructured grids. This is most likely due to the conceptionally more difficult Jacobian computation. Apparently this limitation was not foreseen in the original presentation of the algorithm. Overall, the added robustness makes our method a convenient tool for the structural exploration of large, practical CFD data sets.

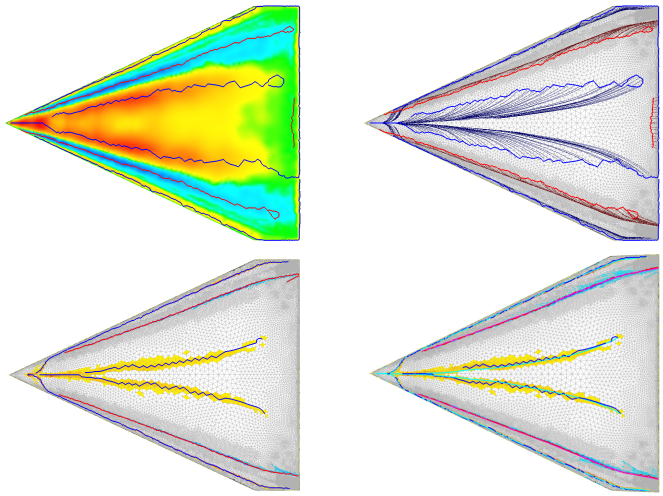


Fig. 4. Delta wing dataset. **Upper left:** Colormap of flow divergence and corresponding ridge lines (positive: blue, negative: red). **Upper right:** Divergence ridge lines and streamlines started from ridge line points. **Lower left:** Colormap of streamline accumulation scalar field (only cells with hits drawn) and corresponding flow driven ridge lines (attachment: blue, separation: red). **Lower right:** Streamline accumulation and flow driven ridge lines together with final results (attachment: pink, separation: cyan).

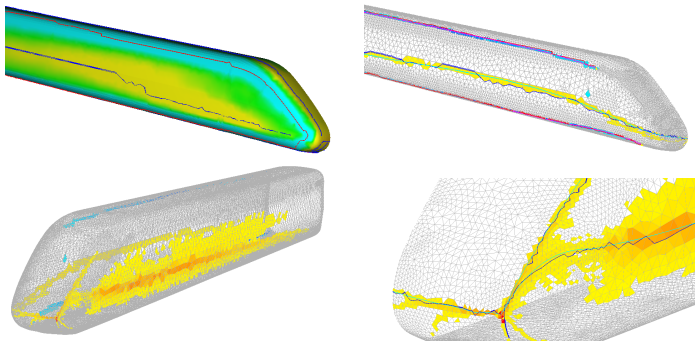


Fig. 5. ICE train dataset. **Upper left:** Colormap of flow divergence and corresponding ridge lines **Upper right:** Colormap of streamline accumulation (only cells with hits drawn), flow driven ridge lines (attachment: red, separation: blue), final results (cyan/pink). **Lower left:** Colormap of streamline accumulation. Note the widely spread accumulation on the train side indicating slow convergence and hence a weak feature. **Lower right:** Zoom of train nose, streamline accumulation, flow driven ridge lines and resulting feature lines.

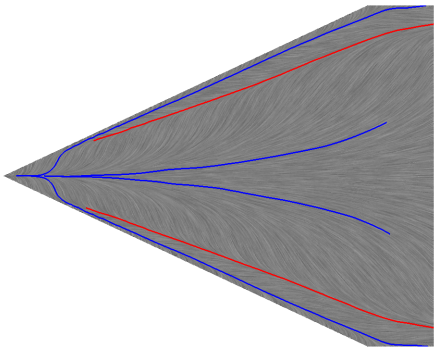


Fig. 6. Resulting features on the delta wing over LIC texture.

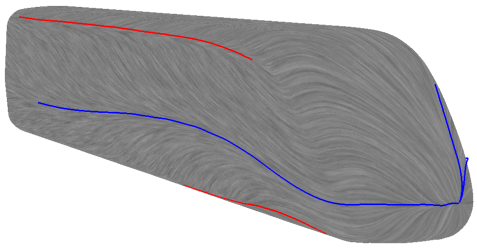


Fig. 7. Resulting features on the ICE train over LIC texture.

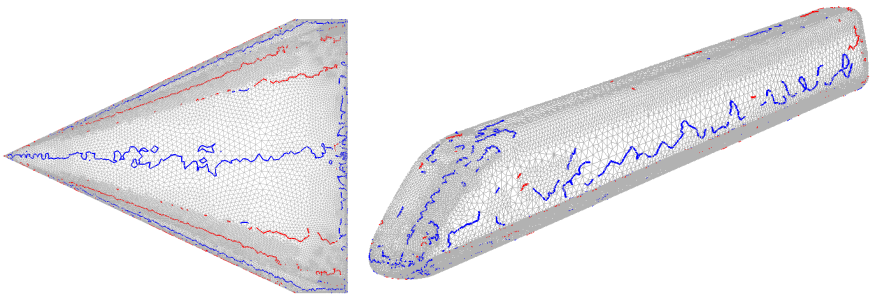


Fig. 8. Results of Kenwright's method, delta wing (left) and ICE train (right).

Acknowledgments

The authors wish to thank Markus Rütten from German Aerospace Center in Göttingen for providing the delta wing and ICE train datasets. Further we thank the members of the FAnToM team at the University of Kaiserslautern and the University of Leipzig for their implementation effort.

References

1. H. Battke, D. Stalling, and H.-C. Hege. Fast line integral convolution for arbitrary surfaces in 3d. In Springer Berlin, editor, *Visualization and Mathematics*, pages 181–195, 1997.
2. B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. *Computer Graphics (SIGGRAPH '93 Proceedings)*, 27(4):263–272, 1993.
3. l. Forssell. Visualizing flow over curvilinear grid surfaces using line integral convolution. In IEEE Computer Society Press, editor, *IEEE Visualization Proceedings*, pages 240–247, Los Alamitos, CA, 1994.
4. A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology if three-dimensional vector fields. In *IEEE Visualization Proceedings*, pages 33 – 40, October 1991.
5. D. N. Kenwright. Automatic detection of open and closed separation and attachment lines. In IEEE Computer Society Press, editor, *IEEE Visualization Proceedings*, pages 151–158, Los Alamitos, CA, 1998.
6. D. N. Kenwright, C. Henze, and C. Levit. Features extraction of separation and attachment lines. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):135–144, 1999.
7. N. Max, R. Crawfis, and C. Grant. Visualizing 3d velocity fields near contour surfaces. In IEEE Computer Society Press, editor, *IEEE Visualization Proceedings*, Los Alamitos, CA, 1994.
8. G. M. Nielson and I.-H. Jung. Tools for computing tangent curves for linearly varying vector fields over tetrahedral domains. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):360–372, 1999.
9. A. Okada and D. L. Kao. Enhanced line integral convolution with flow feature detection. In *Proceedings of IS&T/SPIE Electronic Imaging*, 1997.
10. R. Peikert and M. Roth. The "parallel vectors" operator - a vector field visualization primitive. In *IEEE Visualization Proceedings '00*, pages 263 – 270, 2000.
11. K. Polthier and E. Preuss. Variational approach to vector field decomposition. In *Eurographics Workshop on Scientific Visualization - Preprint No. 448 TU-Berlin, SFB 288*, 2000.
12. K. Polthier and M. Schmies. Straightest geodesics on polyhedral surfaces. pages 391–408, 1998.
13. M. Roth. *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD thesis, ETH Zrich, 2000.
14. A. Sheffer and J. C. Hart. Seamster: Inconspicuous low-distortion texture seam layout. In IEEE Computer Society Press, editor, *IEEE Visualization Proceedings*, pages 291–298, Los Alamitos, CA, 2002.
15. D. Stalling and H.-C. Hege. Fast and resolution independent line integral comvolution. In ACM SIGGRAPH, editor, *Proceedings of SIGGRAPH, Computer Graphics Annual Conference Series*, pages 249–256, 1995.