

# CurVis - Visualizing the Curvature of Vector Fields on the Internet

Holger Theisel and Uwe Rauschenbach  
University of Rostock, Computer Science Department  
PostBox 999, 18051 Rostock, Germany  
{theisel/urausche}@informatik.uni-rostock.de

## Abstract

We compute and color code the curvature of tangent curves of a 2D vector field and apply this as a new visualization technique for flow fields. This way critical points of general topology can clearly be recognized in the visualization. We show that the curvature of tangent curves can be computed in a simple way without using any numerical integration even if a closed formula of the tangent curves themselves does not exist. Furthermore, we introduce CurVis as a system for visualizing the curvature of tangent curves on the Internet. Finally we demonstrate the application of CurVis at an example data set.

**Keywords:** vector field visualization, tangent curves, curvature, vector field topology.

## 1 Introduction

The visualization of vector fields has become one of the main topics in scientific visualization: CFD-data is usually given as vector fields, their visualization may provide new information about many processes in nature, science and technology.

Several techniques for visualizing a vector field have been developed. One of the most important approaches is the visualization of the tangent curves of a vector field. Knowledge of these curves implies knowledge of the directions of the vectors in the vector field. The visualization of tangent curves creates two major problems:

- a) It must be decided how many tangent curves are to be visualized. On the one hand, too many curves lead to a confusing display. On the other hand, there must be as many visualized tangent curves as the user needs to infer the behavior of the remaining curves.
- b) Tangent curves can in general not be described as parametric curves but only as the solution of a system of differential equations. Their visualization requires a numerical solution of those equations. Several approaches for a numerical integration of tangent curves are presented in [1], [4], [5], [6], [7], [8], [10] and [13].

The use of topological concepts is an approach for solving problem a). In [6], [7] and [8] the critical points of the vector field are detected and classified. These points are connected by particular tangent curves, called separation curves. Unfortunately, the classification of the critical points works only for the first order approximation of vector fields. If the first order approximation changes the topology, the method might give us a wrong image of the vector field.

Another approach for solving problem a) can be found in [13]. Here, a line integral convolution technique is used for visualizing the vector field. Since this technique is also based on the numerical integration of tangent curves, the risk of destroying the topology of the vector field remains.

The first approach for dealing with vector fields of general topology can be found in [12]. Here a method is introduced to construct vector fields of general topology using concepts of Clifford algebra.

In the new approach presented in this article we try to use the power of tangent curves but avoid the problems a) and b) described above. We achieve this by visualizing not the tangent curves directly but one of their most characteristic properties: their curvature. The curvature of the tangent curves reflects important properties of the curves, and therefore yields information about the behavior of the entire vector field. Generally, in areas of turbulences we have frequent changes in the flow direction and thus high curvatures of the tangent curves. We show that the visualization of the curvature of tangent curves gives useful additional information about the flow in comparison to known techniques like arrow plots and tangent curve integration.

Flow visualization techniques require powerful computers. Users of personal or portable computers may want to access visualization services located at some distant visualization server over the Internet. The client computers which access such a server will have a variety of display capabilities, and the generated images will have to be transmitted over networks of varying, often low bandwidth. To provide for each client the maximum possible quality of service, an Internet visualization service must be able to adapt to *context parameters* such as display size and bandwidth. We will present the system CurVis, which uses *adaptive image generation* to provide flow visualizations to remote clients over the Internet. CurVis was designed both for establishing the new visualization technique in the Internet and to give the opportunity to compare it with other techniques.

This article is organized in the following way: section 2 shows how to compute the curvature of tangent curves. Based on these results, the new visualization technique is introduced in section 3. The concepts of the CurVis system are introduced in section 4. Section 5 shows an application of CurVis for an example data set.

## 2 Computing the Curvature of Vector Fields

A 2D steady flow is usually described as a 2D vector field  $V(x, y) = (u(x, y), v(x, y))^T$ . A curve  $\mathbf{s} \subseteq \mathbb{R}$  is called a *tangent curve* (stream line) of  $V$  if the following condition is satisfied: For all points  $(x, y) \in \mathbf{s}$ , the tangent vector of the curve in the point  $(x, y)$  has the same direction as the vector  $V(x, y)$ .

For every point  $(x, y) \in \mathbb{R}^2$  there is one and only one tangent curve through it (except for critical points of  $V$ , i.e. points with  $\|V\| = 0$ ). Tangent curves do not intersect each other (except for critical points of  $V$ ). They do not depend on the magnitudes but only on the directions of the vectors in the vector field.

Given a (non-critical) point  $(x_0, y_0)$  in  $V$ , let  $\mathbf{s}$  be the tangent curve through  $(x_0, y_0)$ . Furthermore, let  $\mathbf{s}$  be parametrized in such a way that

$$\mathbf{s}(t_0) = (x_0, y_0) \quad (1)$$

$$\dot{\mathbf{s}}(t_0) = V(\mathbf{s}(x_0, y_0)). \quad (2)$$

( $\dot{\mathbf{s}}(t)$  denotes the tangent vector of  $\mathbf{s}(t)$ ). Then we can compute the second derivative vector  $\ddot{\mathbf{s}}$  of  $\mathbf{s}$  at  $t_0$  by applying the chain rule to (2):

$$\ddot{\mathbf{s}}(t_0) = (u \cdot V_x + v \cdot V_y)(x_0, y_0). \quad (3)$$

Now we can easily compute the signed curvature of  $\mathbf{s}$  in  $(x_0, y_0)$ :

$$\kappa(t_0) = \frac{\det[\dot{\mathbf{s}}(t_0), \ddot{\mathbf{s}}(t_0)]}{\|\dot{\mathbf{s}}(t_0)\|^3}. \quad (4)$$

(2), (3) and (4) have the following consequence: in order to compute the curvature of a tangent curve in a certain point of a vector field it is not necessary to know the tangent curve itself. It is sufficient to know the vector field  $V$  and its partial derivatives.

Inserting (2) and (3) into (4), we obtain a simple formula for the curvature of tangent curves through every point of the vector field:

$$\kappa(V) = \frac{u \cdot \det[V, V_x] + v \cdot \det[V, V_y]}{\|V\|^3}. \quad (5)$$

(5) describes a scalar field in the domain of the vector field  $V$ . This scalar field describes the curvature of the tangent curve in every point of the domain. We call this scalar field  $\kappa(V)$  the *curvature of the vector field  $V$* .  $\kappa(V)$  is only defined for non-critical points. It does not depend on the magnitudes of the vectors in  $V$ .

The *perpendicular vector field*  $V^\perp$  of a 2D vector field  $V = (u, v)^T$  is defined as  $V^\perp := (-v, u)^T$ . For every point of the vector field, the vectors of  $V$  and  $V^\perp$  are perpendicular to each other. We obtain for the curvature of  $V^\perp$ :

$$\kappa(V^\perp) = \frac{u \cdot \det[V, V_y] - v \cdot \det[V, V_x]}{\|V\|^3}. \quad (6)$$

### 3 Visualizing the Curvature of Vector Fields

We want to visualize the curvature  $\kappa$  of a 2D vector field in the following way: compute  $\kappa$  for every point of the domain and color code these values. To do this we use a continuous color coding map with the following properties: a negative value is mapped to a green color, a positive value is mapped to a red color. The higher the magnitude of the value

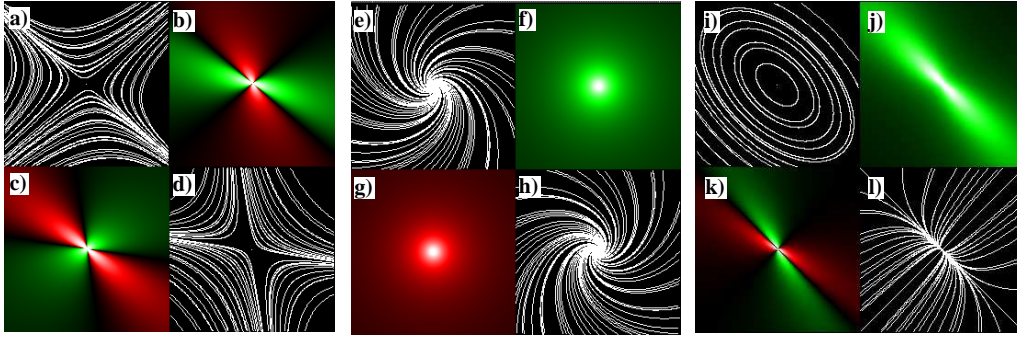


Figure 1: Linear vector field with saddle point (a..d); linear vector field with repelling focus (e..h); linear vector field with center (i..l)

the lighter the color gets. A zero value gives black; if the value diverges to plus (minus) infinity the red (green) color tends to white.

The pictures a-d of figure 1 give an example of the vector field  $V(x, y) = \begin{pmatrix} 1 \\ 3 \end{pmatrix} x + \begin{pmatrix} 4 \\ 1 \end{pmatrix} y$ . This linear vector field has a critical point at  $(0, 0)$  - a saddle point. Figure 1a shows a numerical tangent curve integration. Figure 1b is the visualization of its curvature. Figures 1d and 1c show the same for the perpendicular vector field  $V^\perp$ . In this case,  $V^\perp$  has a saddle point at  $(0, 0)$  as well. Note that generally the topology of a vector field and its perpendicular vector field might differ.

The reason for visualizing the curvature of both  $V$  and  $V^\perp$  is shown by considering the following visualization properties:

In the curvature visualization b) of figure 1 the critical point appears as highlight. Considering (5),  $\kappa(V)$  tends to infinity only if the denominator of  $\kappa$  tends to 0. This occurs only at critical points. Therefore, a highlight in the curvature visualization always indicates a critical point in the vector field. The reverse question arises: does every critical point produce a highlight in the curvature visualizations? The answer is yes, if we exclude certain degenerate points. A *degenerate critical point* of a vector field  $V$  is a critical point where the directions of the vectors of  $V$  do not change in the neighborhood of the critical point. For non-degenerate critical points, we have the following

**Theorem 1** *In the neighborhood of a non-degenerate critical point of a 2D vector field  $V$ , the curvature of  $V$  or  $V^\perp$  (or both curvatures) tend to infinity.*

An exact definition of a degenerate critical point and the proof of this theorem can be found in [14]. The same theorem can be formulated in the following way: non-degenerate critical points in a vector field  $V$  always produce highlights in the visualization of the curvature of  $V$  or  $V^\perp$ .

Considering the curvature visualizations b) and c) of figure 1 again, another question arises: Do the curvature visualizations of  $V$  and  $V^\perp$  contain all information of  $V$ ? The answer is given by

**Theorem 2** *Given are two 2D vector fields  $V_1$  and  $V_2$  which have non-constant direction fields. If  $\kappa(V_1) = \kappa(V_2)$  and  $\kappa(V_1^\perp) = \kappa(V_2^\perp)$  then the directions of the vectors of  $V_1$  and  $V_2$  coincide in every point.*

See [14] for a proof. Theorem 2 has an interesting consequence: the curvatures of  $V$  and  $V^\perp$  together contain all information about the directions of the vectors in  $V$ . Therefore, the curvatures of  $V$  and  $V^\perp$  contain all information about the topology of  $V$ . This statement is true for vector fields of general topology.

Pictures e-h of figure 1 show a linear vector field with a repelling focus. Figure 1e is the numerical stream line integration, figure 1f is the curvature visualization. Figures 1h and 1g show the same for the perpendicular vector field. The repelling node appears completely green around the highlight in the curvature visualization and completely red in the curvature visualization of the perpendicular vector field. Figures 1 i-l show the visualization of a center. It appears completely green around the highlight in the curvature visualization (figure 1j) and has 4 different areas (colored red or green) each of 90 degrees in the perpendicular curvature visualization (figure 1k).

All critical points of figure 1 are of first order, i.e. they have  $\det[V_x, V_y] \neq 0$  in the critical points. They can therefore be classified using the topology concepts described in [6]. Figure 1 also shows that the different kinds of critical points appear differently in the curvature visualizations. A detailed description of how to classify a first order critical point from the curvature visualization can be found in [15].

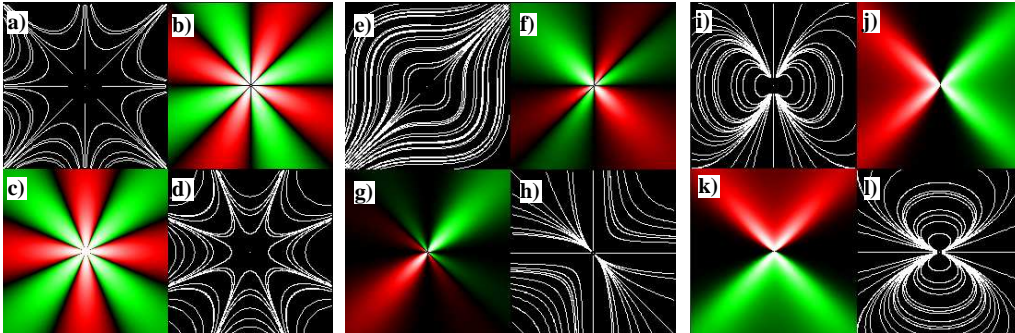


Figure 2: Higher order saddle point (a..d); critical point with two elliptic sectors (e..h); dipole (i..l)

Figure 2 shows a collection of higher order critical points, i.e. points with  $\det[V_x, V_y] = 0$ . None of these points can be treated using the topology methods of [6] but their curvature visualization gives a fairly good impression of them. Figures 2 a-d show a saddle point with 4 pairs of tangent curves through it. In the curvature visualization (figure 2b) we have eight differently colored sections around the critical point. The perpendicular field (figure 2c) has eight different sections as well. Figure 2 e-h shows the visualization of the vector field  $V(x, y) = (y^2, x^2)^T$  in the range  $[-1, 1] \times [-1, 1]$ . This vector field has a critical point with two elliptic sections in  $(0, 0)$ . Observing the stream line integration (figure 2e), this critical point may be missed. The curvature visualization (figure 2f) shows it clearly as a highlight with six differently colored sections around it. Here the visualization of the perpendicular curvature has two differently colored areas

(figure 2g). Figures 2 i-l show the visualization of a vector field describing a dipole. Both the visualization of its curvature (figure 2j) and its perpendicular curvature (figure 2k) show two differently colored sections around the highlighted critical point.

A general algorithm which infers the topology of higher order critical points from the curvature visualizations is still unknown. Nevertheless, the higher order critical points of figure 2 can be well distinguished from the first order critical points of figure 1 by their curvature visualizations.

## 4 The CurVis System

We developed the CurVis system not only for establishing our visualization technique on the Internet but also in order to compare it with known techniques like arrow plots and stream line integration. A number of problems concerning adaptive image generation had to be solved.

### 4.1 Adaptive Image Generation

Adaptive image generation can stretch the limits imposed by low bandwidth, restricted client display resources and processing power. It can save resources by generating images on the fly at a resolution and with a number of colors which can be displayed at client side. Compression of the generated image using the right compression method further decreases the bandwidth demands. A method which supports progressive refinement of the whole image (e.g., progressive JPEG) or even only of selected regions of interest (see [11]) allows the user to early assess the value of the visualization for his/her goals and to cancel unnecessary data transmissions. The adaptive image generation process is controlled by a set of *context parameters* (see figure 3):

- Display context: describes the client display size and possible number of colors
- Technique: selects the visualization technique
- Zooming: describes the mapping from the grid underlying the data to the grid underlying the pixel image to be generated
- Network: describes the available bandwidth

Depending on these context parameters, a visualization technique can be selected and parameterized by the system such that the image fits the client display capabilities.

### 4.2 System Overview

The system CurVis provides three visualization techniques: the classic streamlines, a simple arrow plot and the curvature plot technique presented in section 3. Furthermore, the critical points of the vector field can be analyzed using the approach presented in [6] and transmitted as a textual description. Each visualization module creates a bitmap image. More visualization modules can easily be added, as each module runs as a CGI program on the server. This allows the simple integration of arbitrary executables as long as they obey some basic interface specifications. When the image is generated, the

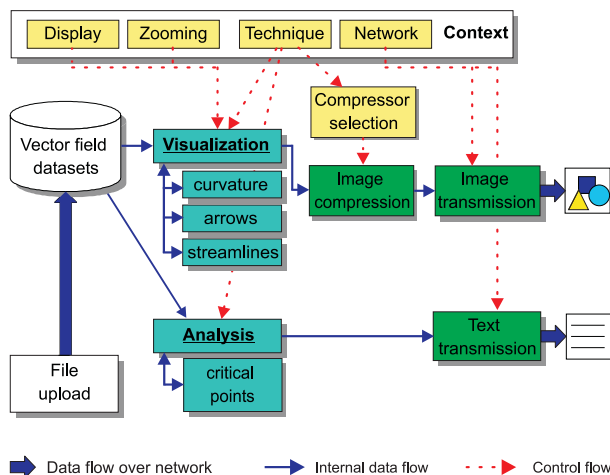


Figure 3: Block diagram of CurVis

context parameters discussed above are considered. An already generated image can be zoomed in by altering the zooming factor in the context parameter set. In this case, a new image of higher resolution is generated using the new context. Remote users can upload their data files to the CurVis server for visualization. CurVis automatically selects the image compression method best suited to the visualization generated. Figure 3 shows a block diagram of CurVis.

### 4.3 Compression Method Selection

CurVis uses image compression methods in order to save transmission bandwidth. The images produced by the different visualization techniques require different compression methods. One might think that the with-loss compression algorithm JPEG is always superior over loss-free algorithms - but that is not the case. As JPEG has been developed for continuous-tone true color images, it delivers best compression on this image class. On images with 256 or less colors, loss-free compressors like PNG or GIF are often superior to JPEG. Figure 4 illustrates this fact. Six images were color quantized to three up to nine bit planes. The resulting images were compressed using PNG on the one hand and JPEG with different quality parameter values<sup>1</sup> on the other hand. The curves in figure 4 represent the average compression ratio for the six images. Furthermore, two curves have been added depicting the compression ratio of the images that are best and worst compressible using PNG. It can easily be seen that the compression ratio of JPEG is independent from the number of colors, and that for images with only a few colors the loss-free PNG is superior over the with-loss JPEG. However, the break-even point between JPEG and PNG cannot be derived from the graph for *arbitrary images*, as the variation between individual images is quite high. Instead, test compressions are necessary.

With CurVis, we are in a better situation. Since the different techniques generate images with very different characteristics, we are able to select the compression method

<sup>1</sup>The quality parameter controls the quantization and thus the image quality. There is however, no direct relation to subjective quality measures like MOS or objective quality measures like PSNR.

best suited using our knowledge about these characteristics. Arrow plots and streamline images have only a few colors and large areas shaded in the background color, making them candidates for loss-free compression using PNG or GIF<sup>2</sup>. Curvature plots, showing smooth gradients of color, can be better compressed using JPEG. Preferably, the progressive JPEG mode is used. Figure 5 provides a comparison. Using JPEG, a streamline image can be compressed to the size of its GIF equivalent only by applying strong quantization which leads to very poor image quality.

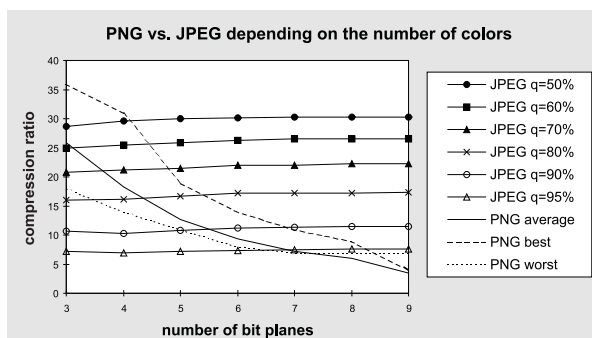


Figure 4: Compression ratio of PNG and JPEG depending on the number of colors

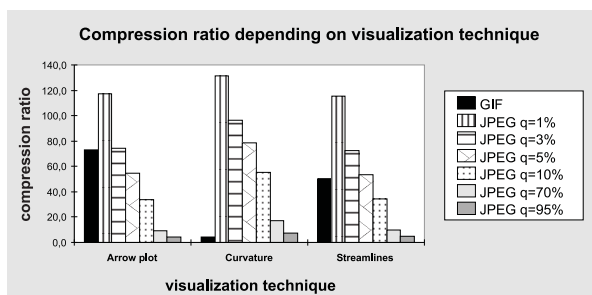


Figure 5: Suitability of GIF and JPEG for the compression of different flow visualizations

## 5 Applications and Results

We applied CurVis to a test data set which describes the flow of water in the bay area of the Baltic Sea near Greifswald, Germany (Greifswalder Bodden). The bay covers an area of  $23 \times 26$  km. The maximal depth of the water is 12 meters. The flow in this shallow water can be considered as a 2D flow. The vectors of the sample points on a regular  $115 \times 103$  grid have been obtained by a numerical simulation. Between the grid points a bilinear interpolation is applied.

Figure 6 (left) shows the CurVis start page for the test data set. Here the system is configured to deliver the curvature and perpendicular curvature visualization of the flow. Figure 6 (right) shows the system answer of figure 6 (left). For comparison, we

<sup>2</sup>We are using GIF instead of PNG in CurVis because nearly all web browsers can display it.



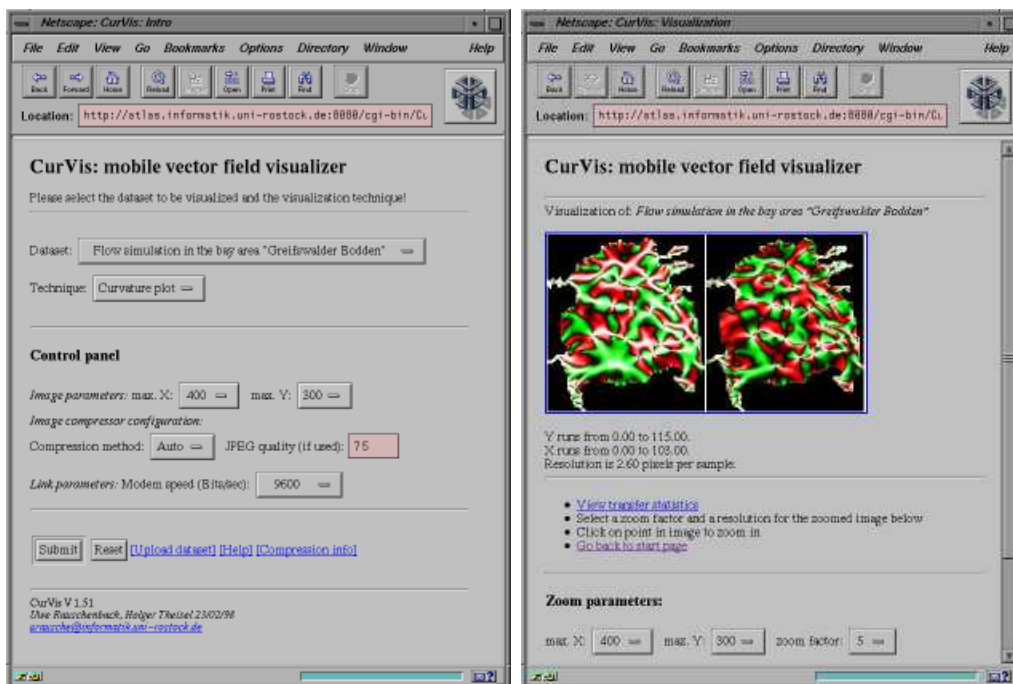


Figure 6: CurVis - start page (left), curvature visualization (right)

computed the curvature visualization (figure 7 b and c) as well as a numerical streamline integration (figure 7a) using Curvis.

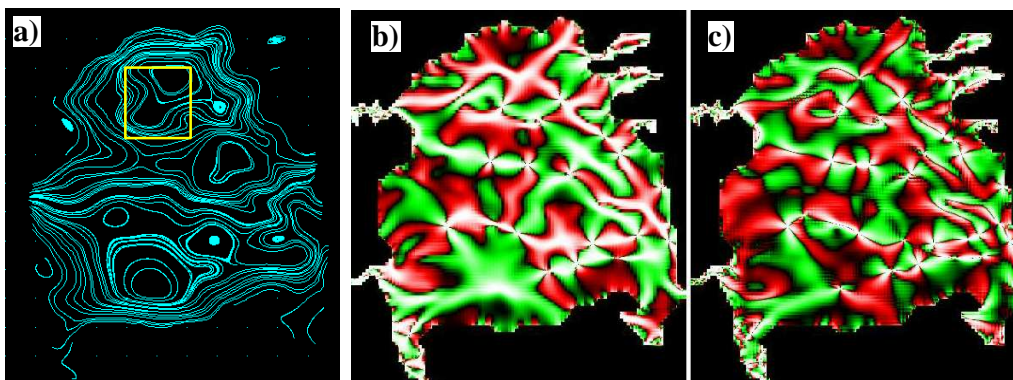


Figure 7: Test data set; a: numerical streamline integration, b: curvature visualization, c: perpendicular curvature visualization

Figures 7b and 7c give an overview about number and location of critical points in the flow. In contrary to the streamline integration of figure 7a, there is no danger of missing a critical point.

Figure 8 shows a zoom into the marked area of figure 7a. Again, figure 8a shows the streamline integration image, figure 8b is the curvature visualization, and figure 8c is the perpendicular curvature visualization. In the zoomed area of the flow we have 3 critical points - two centers and a saddle point. The centers appear completely red around the highlights in the curvature visualization. In the perpendicular curvature visualization

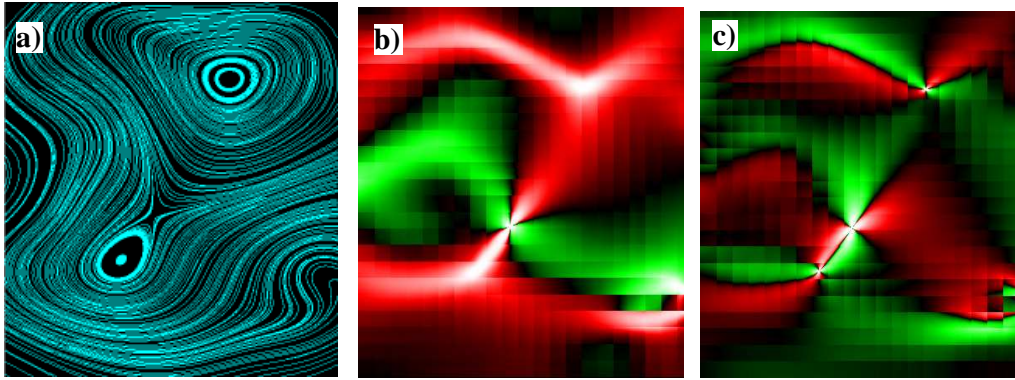


Figure 8: Test data set (magnification); a: numerical streamline integration, b: curvature visualization, c: perpendicular curvature visualization

they have 4 different colored sections of each 90 degree around the highlight. The saddle point has 4 different colored sections around the highlighted critical point in both the visualization of curvature and perpendicular curvature.

Furthermore, figures 8b and 8c show clearly the underlying grid structure of the bilinear vector field. This means that a bilinearly interpolated vector field has no curvature continuous tangent curves across the boundaries of the grid cells.

## 6 Conclusions

We introduced a technique for visualizing 2D vector fields by computing and color coding the curvature of the tangent curves. This visualization is "exact", we do not have to apply any numerical solution methods. Non-degenerate critical points can clearly be recognized as highlights in the curvature visualization. The visualization of the curvature of  $V$  and  $V^\perp$  contains all information about the topology of  $V$ .

However, there are drawbacks to the technique as well. The curvature plot does not directly provide information about the direction of the vector in a point – which might be useful for some applications. A combination of the curvature visualization and a numerical stream line integration (as shown in figures 7 and 8) can overcome this.

For the future we intend to visualize the curvature of 3D vector fields. Here suitable techniques of volume visualization have to be applied.

CurVis can be accessed at  
<http://www.informatik.uni-rostock.de/Projekte/movi/IIS/curvisrdr.html>.

## 7 Acknowledgements

The authors thank Prof. Gerald Farin from Arizona State University and Prof. Heidrun Schumann from Rostock University for their constant support and encouragement. The

authors also thank Dr. Kurt Frischmut from Rostock University for providing the test data set. Parts of the work presented have been supported by the German Science Foundation (Deutsche Forschungsgemeinschaft) under contract no. Schu-887/3-1.

## References

- [1] S. Bryson and C. Levit. The Virtual Wind Tunnel. *IEEE Computer Graphics and Applications*, 12(4), pp. 25-34, July 1992.
- [2] H. Davis. *Introduction to vector analysis*. Allyn and Bacon, Inc., Boston, 1967.
- [3] P.A. Firby and C.F. Gardiner. *Surface Topology*. Ellis Horwood, Chichester, 1982.
- [4] T. Frühauf. Interactive Visualization of Vector Data in Unstructured Volumes. *Computers and Graphics*, 18(1), pp. 73-80, 1994
- [5] T. Frühauf. Raycasting of Nonregularly Structured Volume Data. *Proceedings Eurographics '94*, pp. 293-303, 1994.
- [6] J.L. Helman and L. Hesselink. Representation and Display of Vector Field Topology in Fluid Data Sets. *IEEE Computer*, Vol. 22, No. 8, Aug. 1989, pp. 27-36.
- [7] J.L. Helman and L. Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, Vol. 11, May 1991, pp. 36-46.
- [8] I.H. Jung. *Topological Visualization Method of Vector Fields in Fluid Data Sets*. Master's thesis, Dept. of Computer Science, Arizona State University, USA, 1993.
- [9] W. de Leeuw and J. van Wijk. A Probe for Local Flow Field Visualization. *Proceedings IEEE Visualization '93*, pp. 39-45, 1993.
- [10] F. Post and T. van Walsum. Fluid Flow Visualization. In H. Hagen, H. Müller and G.M. Nielson, editors, *Focus on Scientific Visualization*, Springer-Verlag Berlin, pp. 1-40, 1993.
- [11] U. Rauschenbach. Progressive Image Transmission using Levels of Detail and Regions of Interest. *Proceedings IASTED Conf. on Computer Graphics and Imaging - CGIM'98*, Halifax, Nova Scotia, Canada, June 1-4, 1998.
- [12] G. Scheuermann, H. Hagen, H. Krüger, M. Menzel and A. Rockwood. *Visualization of Higher Order Singularities in Vector Fields*. Proc. of IEEE Visualization '97, IEEE Computer Society, Los Alamitos, CA, 1997, pp. 67-74, 1997.
- [13] D. Stalling and H. Hege. Fast and Resolution Independent Line Integral Convolution. *Proceedings Siggraph '95*, Los Angeles, pp. 249-256, 1995.
- [14] H. Theisel. *Vector Field Curvature and Applications*. PhD thesis, Dept. of Computer Science, University Rostock, Germany, 1996,  
also available at <http://www.icg.informatik.uni-rostock.de/~theisel/> .
- [15] H. Theisel. *Obtaining the Topology of Bilinear Vector Fields from their Curvature Visualizations*. Rostocker Informatik-Berichte 21, ISSN 0233-0784, 1998,  
also available at <http://www.icg.informatik.uni-rostock.de/~theisel/> .