Line Integral Convolution on Triangulated Surfaces

Christian Teitzel, Roberto Grosso and Thomas Ertl
Lehrstuhl für Graphische Datenverarbeitung (IMMD IX)
Universität Erlangen-Nürnberg
{teitzel,grosso,ertl}@informatik.uni-erlangen.de
http://www9.informatik.uni-erlangen.de/

Abstract

Line Integral Convolution (LIC), introduced by Cabral and Leedom in 1993, is a powerful technique for generating striking images of vector data. Based on local filtering of an input texture along a curved stream line segment in a vector field, it is possible to depict directional information of the vector field at pixel resolution.

The methods suggested so far can handle structured grids only. Now we present an approach that works both on two-dimensional unstructured grids and directly on triangulated surfaces in three-dimensional space. Because unstructured meshes often occur in real applications, this feature makes LIC available for a number of new applications.

1 Introduction

Imaging vector fields has a lot of applications in scientific visualization, as well as in image processing and in special effects. In 1993 Cabral and Leedom introduced a method for computing textured images from vector fields using line integral convolution (LIC) in [1], and later Forssell [4], Stalling and Hege [13] presented a number of improvements for generating LIC images. The advantage of striking images is their high spatial resolution. However, the approaches mentioned above are restricted to vector data given on structured grids. Since unstructured meshes often occur in scientific simulation, our aim was to develop and implement an algorithm for creating striking images on two-dimensional unstructured grids and also on two-dimensional manifolds in three-dimensional space.

Vector fields can be visualized in a number of different manners. Traditionally, vector data are visualized by simple arrow plots. More advanced methods use icons [10] that show some more parameters of the field. However, these icons are sometimes difficult to interpret. Sophisticated approaches include methods like stream lines [7], stream surfaces [9], volume flows [11] and various particle tracing techniques to depict properties of vector fields. These approaches are restricted to a rather coarse spatial resolution. In contrast to them, texture-based methods achieve a much higher resolution.

In an early texture-based method, introduced by van Wijk [14] in 1991, a texture with white noise is convolved along a straight line segment oriented parallel to the local vector direction. In 1993 Cabral and Leedom [1] introduced a modification of this method called line integral convolution (LIC). In their algorithm convolution takes place along curved stream line segments. In 1995 Stalling and Hege [13] made LIC much faster, more accurate and independent of resolution.

In 1994 Forssell [4] presented an extension that allows to map flat LIC images onto curvilinear surfaces in three dimensions. A problem of this method is the distortion of length during the mapping process. In this paper we are presenting an algorithm for generating striking images that operates on triangles and allows computing LIC images directly on a surface in three-dimensional space without mapping.

2 Basics of Line Integral Convolution

The LIC algorithm filters an input image along stream lines, also denoted integral curves or flow lines, of a given vector field and generates a texture as output. In most cases in scientific visualization a texture with white noise is used as input image.

The Intensity I of an output texture pixel located at $x_0 = \sigma(s_0)$ is given by

$$I(x_0) = \int_{s_0-L}^{s_0+L} k(s - s_0)\, T(\sigma(s))\, ds\,,$$

where $\sigma(s)$ denotes a stream line of the vector field parameterized by arclength, T the intensity of the input texture and k a filter kernel. If we choose a constant filter kernel k and consider that T is constant at each pixel, the convolution integral can be computed by sampling the input texture T at locations x_i along the stream line $\sigma(s)$:

$$I(x_0) = k \sum_{i=-n}^{n} T(x_i)\,,$$

where we choose $k = 1/(2n + 1)$ to normalize the intensity.

The convolution causes pixel intensities to be highly correlated along individual stream lines but independent in directions perpendicular to them. In the resulting images the stream lines are clearly visible.

3 Line Integral Convolution on Triangles

The LIC algorithms presented up to now can only visualize vector fields given on structured grids via striking images. We now introduce an approach that works both on two-dimensional unstructured grids and directly on triangulated surfaces in three-dimensional space. Therefore LIC is now available for a number of new applications.

We have implemented our algorithm in the C++ programming language as an IRIS Explorer module [6]. Our LIC module can handle vector fields on structured and unstructured grids. The vector fields are given to the module in a cxLattice or cxPyramid IRIS Explorer structure respectively.

Due to the modules for multiblock data [5] which were recently developed at our institute and implemented within the framework of the IRIS Explorer visualization environment, we are able to compute LIC images for vector fields on structured and unstructured multiblock grids. Furthermore it is possible to distribute the LIC processes of different blocks to different machines or processors to save computing time (see figure 1).

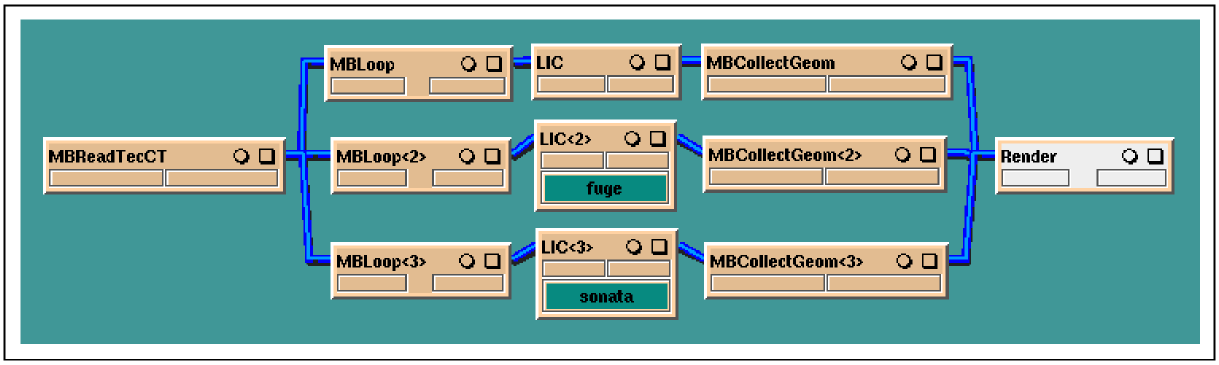

Figure 1: IRIS Explorer map with three LIC modules running on three different machines. The MBLoop and MBCollectGeom modules are used to handle the multiblock data set.

3.1 Texture on Unstructured Grids

The purpose of this subsection is to show how to define textures on unstructured grids. In case of structured meshes a cell in computing space (c-space) is a square. Therefore we can easily map a single texture pixel (texel) or an $m \times n$ matrix of texels to a cell in c-space. And so we can map the whole texture to the entire grid. Since in unstructured meshes a cell is a triangle, we have to find a new algorithm to generate textures on triangles.

First of all let us look at the output texture and imagine that a white noise as input is already given on the entire grid. Remark that the vector field is known at the vertices of the triangles and can be calculated at interior points by barycentric combination.

Trying to define a single texture on the complete unstructured grid causes much trouble. Hence we have decided to endow each triangle of the mesh with its own small texture. In the user interface of the LIC module you can set the size of these local output textures, and then one half plus the diagonal of the local output texture square is mapped to each triangle (compare figure 2). Now for all texels on the triangles the intensity is computed by convolving a stochastical input function along a curved stream line segment through this texel.

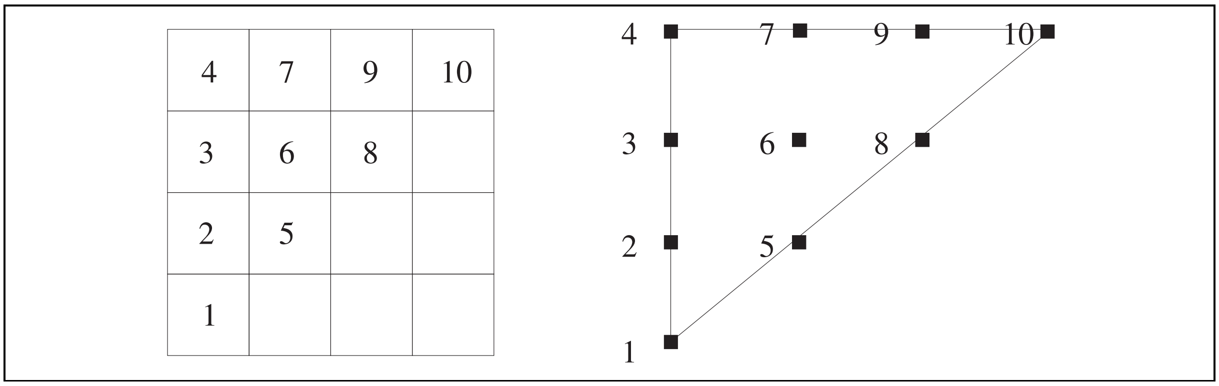

Figure 2: Correlation between output texture and triangle.

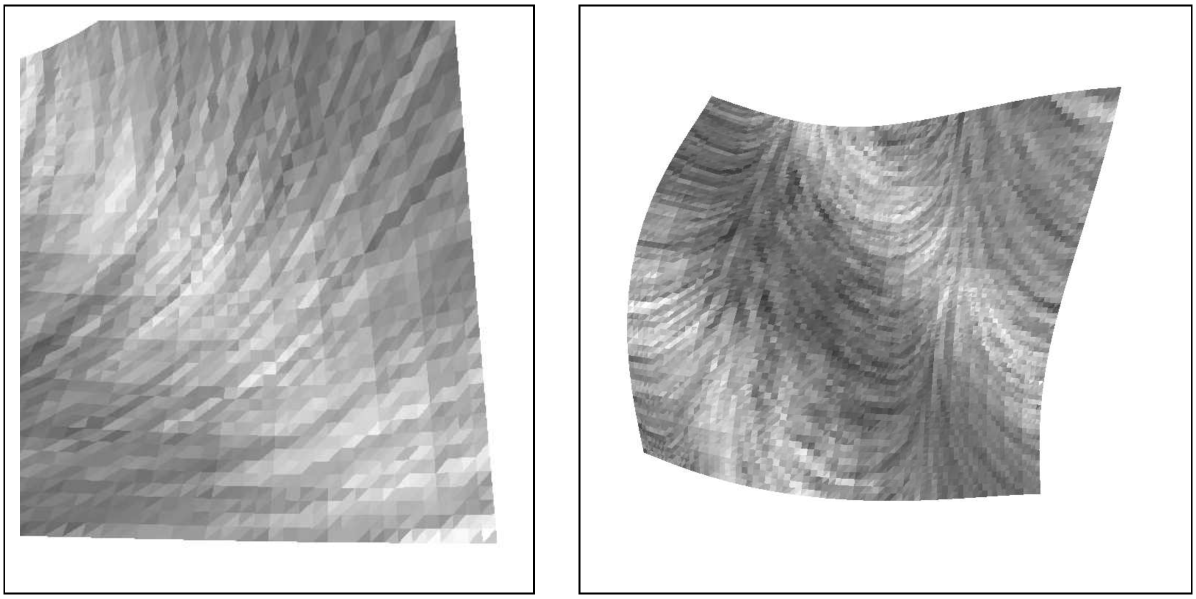

Figure 3: LIC texture with one colour per triangle on triangulated surfaces in three-dimensional space.

Now we discuss the stochastical input function. In the LIC module we have implemented three different kinds of white noise input.

The first called *texture* is similar to the way we create the output texture. A small texture with white noise is generated by a random mechanism. Of course the user can specify the size of this stochastical texture. After that the half of this texture is mapped to each triangle of the grid. Thus we do not get really white noise but a pattern as input. Nevertheless, the results of this method are quite good, as you can see in figure 7.

The second technique is named *interpolation*. Here each vertex of the unstructured mesh gets a random grey value. Therefore we obtain a stochastical input on the whole grid. The grey value at an interior point of a triangle is interpolated. In this mode we have not any longer a texture as stochastical input. The pictures computed with this algorithm seem to be smoother than that ones obtained by the *texture* method (see figure 8 and 9).

The last treatment is the simplest one. Each triangle randomly gets a single grey value. In this approach also the output texture shrinks to one texel per triangle (see figure 3). Thus we obtain again a stochastical input on the whole grid. To obtain an image with a high spatial resolution you can refine the triangles. At the moment every triangle is subdivided into four triangles if you choose the next refinement level.

3.2 Tracing on Triangulated Surfaces in Three-Dimensional Space

If we want to compute an LIC image on a triangulated surface in three-dimensional space, we have to compute the stream lines on the triangles. Then we can apply the flat LIC algorithm mentioned in section 2. For computing stream lines traditional Runge-Kutta methods can be used. More sophisticated and much faster are embedded Runge-Kutta methods with error monitoring and adaptive step size control [3]. In many applications vector fields arise that are very rough, so that higher order algorithms like extrapolation methods are useless.

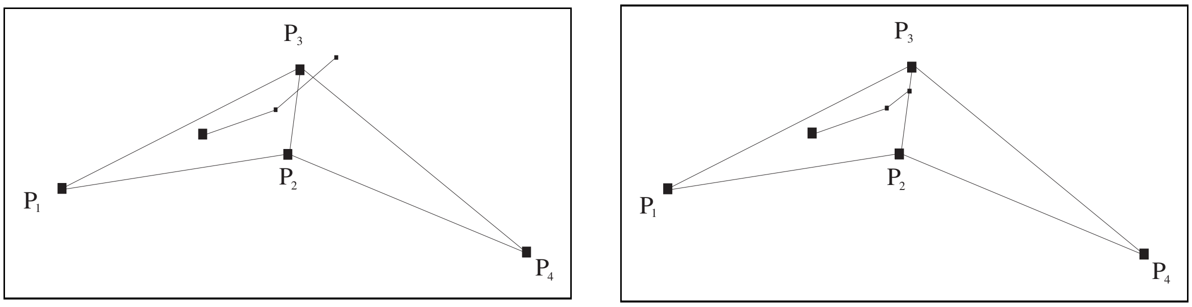

Figure 4: Clipping the integration path at edges. In this example situation the integration path starts in triangle 1 with the vertices P_1, P_2 and P_3. During the second integration step the path hits the boundary between triangle 1 and triangle 2. At this edge the path is clipped to prevent it from leaving the surface.

Before we can integrate the vector field on a none planar triangulated surface, we have to project it onto the triangle where the integration begins. Then stream line computing can start. If the stream line hits the triangle boundary, it has to be clipped to prevent it from leaving the surface (see figure 4). After that the computation restarts at the clipping point in the neighbour triangle.

A problem occurs if the interpolated vector at an edge lies in the critical region shown in figure 5. Then the projection onto triangle 1, as well as that onto triangle 2 points back into the other triangle. To solve this problem at least partially we have introduced an intermediate plane. The normal of this plane is the sum of the normals of the adjacent triangles (see figure 6). Now a vector at an edge is projected first to the intermediate plane, and then this projection is projected to both triangles. If the vector at the edge lies in the right half of the critical region, both projections point forward now, and the integration can go on. If the vector at the edge lies in the left half of the critical region, both projections point backward, and the integration of this particular stream line comes to an end. In such a case the output pixel intensity at this point is not correlated along the stream lines. The same applies also to a neighbourhood along the edge because of the continuity of the vector interpolation. This causes that such edges are clearly visible. See the sphere in figure 7. At the vertices there are roughly the same problems and solutions.

For solving this problem completely it is necessary to do the integration on a C^1 differentiable surface. An idea is to compute a differentiable spline surface from the mesh data via a split surface approach [8], and then integrate the vector field on this surface. Possible approaches could be Clough-Tocher- [2] or Powell-Sabin-interpolation [12]. A remaining difficulty is that both interpolation methods need the first derivatives at the

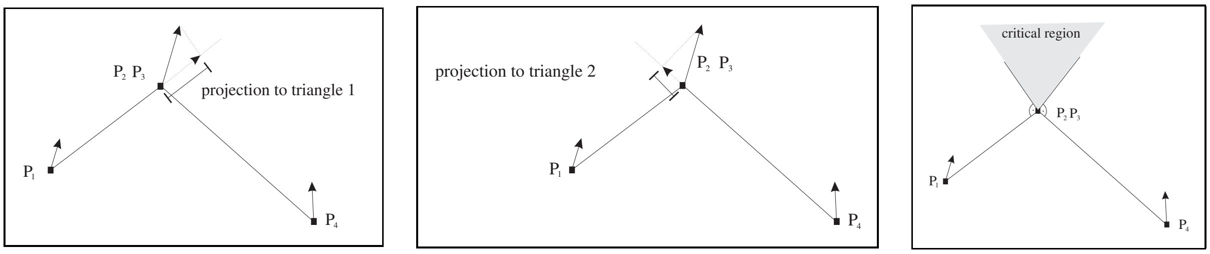

Figure 5: Problems at edges.

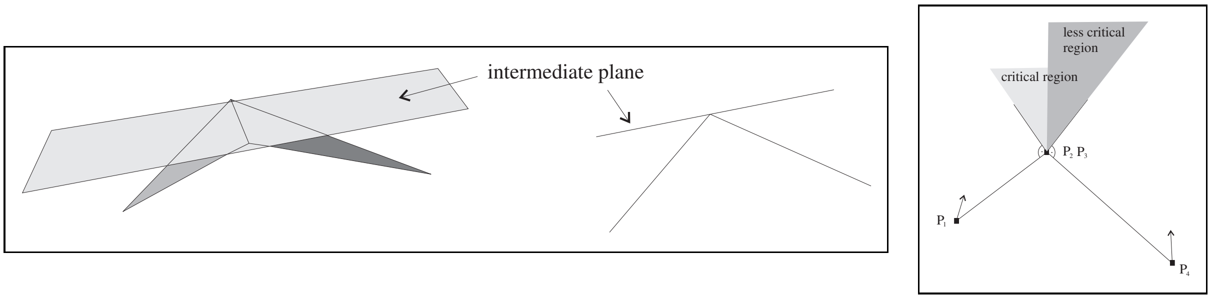

Figure 6: Intermediate plane to shrink the critical region.

vertices of the grid. The Clough-Tocher-interpolation even requires first derivatives at the edge midpoints transverse to the edges. However, these derivatives correspond to our intermediate planes at edges and vertices. So the problem could be solved.

4 Applications

Together with our modules for multiblock data [5], The new LIC module opens up a lot of new applications since unstructured multiblock grids are of increasing interest in computer graphics, as well as in fluid mechanics and material science. Especially the high spatial resolution of striking images is of great interest.

In cooperation with the Lehrstuhl für Strömungsmechanik (LSTM) of the University of Erlangen, visualization is used as a tool for optimal design of a stirrer for sewage purification plants. During the process of development, numerically computed data has been compared with data arising out of experimental measurements by means of computer graphics. The multiblock data set shown in figure 8 arises out of measurements. There the input intensity is received by interpolating the randomly given intensities at the nodes of the mesh.

A second collaboration with the LSTM deals with air bag simulation (see figure 9). This data set consists of thirteen blocks and shows a slice through an air bag of a car. The velocity field is visualized during the initial ignition of the air bag. Again the interpolation method has been used to compute the input intensities.

In another cooperation with the Lehrstuhl für Werkstoffwissenschaften VI of the University of Erlangen, the velocity field is visualized in a Czochralski furnace for crystal growing. See the picture on the left hand side of figure 7. This LIC image has been produced with a white noise texture on each triangle as input.

In all examples mentioned above the vector fields are given on curvilinear grids. Furthermore, the data sets of the collaborations with the LSTM are multiblock data sets. Together with our modules for multiblock data (compare figure 1 and [5]) we can control a particular block or a subsequence of blocks. Thus we are able to choose different texture sizes or different levels of refinement for different blocks. The striking images in figure 8 and 9 have been generated with small texture sizes in the small blocks and with large texture sizes in the big ones. Even various input texture methods can be used in different blocks. As mentioned earlier it is also possible to distribute the LIC modules to several machines.

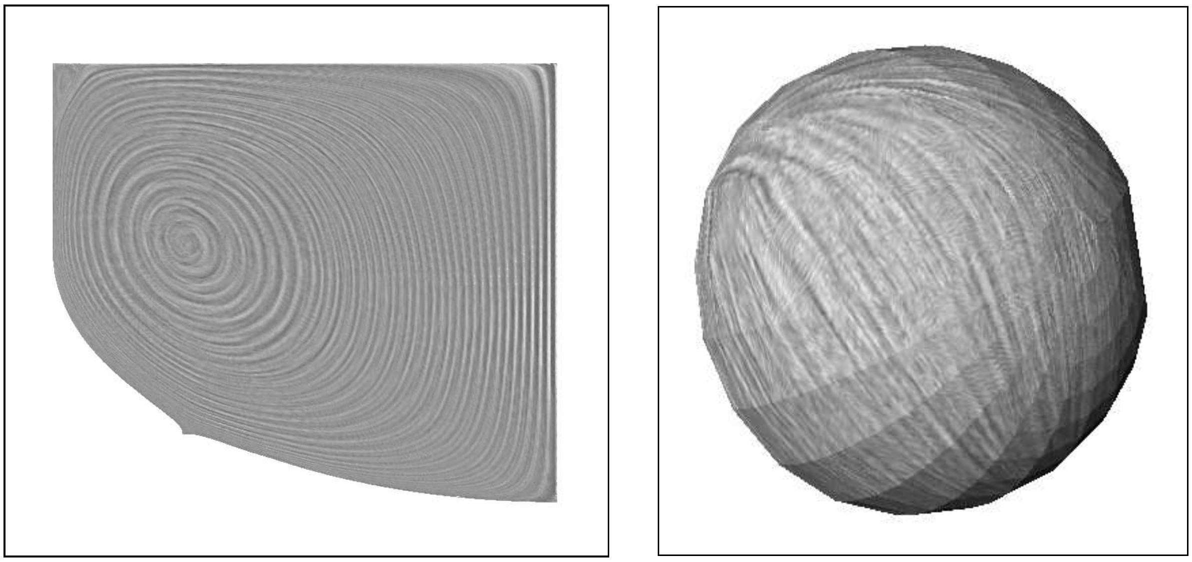

Figure 7: On the left hand side the stream lines of the velocity field in a Czochralski furnace for crystal growing are visualized. On the right a vector field on a sphere is visualized. Both LIC images have been generated with a white noise texture on each triangle as input.

5 Conclusion

We have introduced new techniques for line integral convolution which allow to generate striking images on unstructured meshes and on triangulated surfaces in three-dimensional space. Our algorithm is implemented within the framework of the IRIS Explorer visualization environment. Due to the modules for multiblock data which we recently developed we are able to compute textured images for vector fields on structured and unstructured multiblock grids. Besides, we are able to distribute the LIC processes of different blocks to different machines or processors to accelerate image creating.

The new methods open up a number of new applications since unstructured grids are more and more important to researchers. The production of striking images is of growing interest in computer graphics, as well as in engineering, especially in fluid mechanics and material science.

There are several directions of future research. First of all we intend to think about adaptive methods for choosing the right texture size or for refining the triangles in the *single colour per triangle* algorithm. The number of texels should not be fix at each triangle but should be chosen depending on the size of the triangles. In roughly the same way the triangle refinement should work in the *single colour per triangle* method.

A second improvement would be a solution for the problem resulting from the projections at the edges and vertices. The idea mentioned above is to compute a differentiable spline surface from the grid data via an interpolation approach from Clough and Tocher or Powell and Sabin, and then integrate the vector field on this surface.

Another aspect is the extension of the LIC algorithm to tetrahedra and so to three-dimensional unstructured meshes. Finally, there is the possibility to take advantage of the hardware convolution capability using the texture mapping subsystem of current SiliconGraphics workstations if we compute LIC images on two-dimensional surfaces.

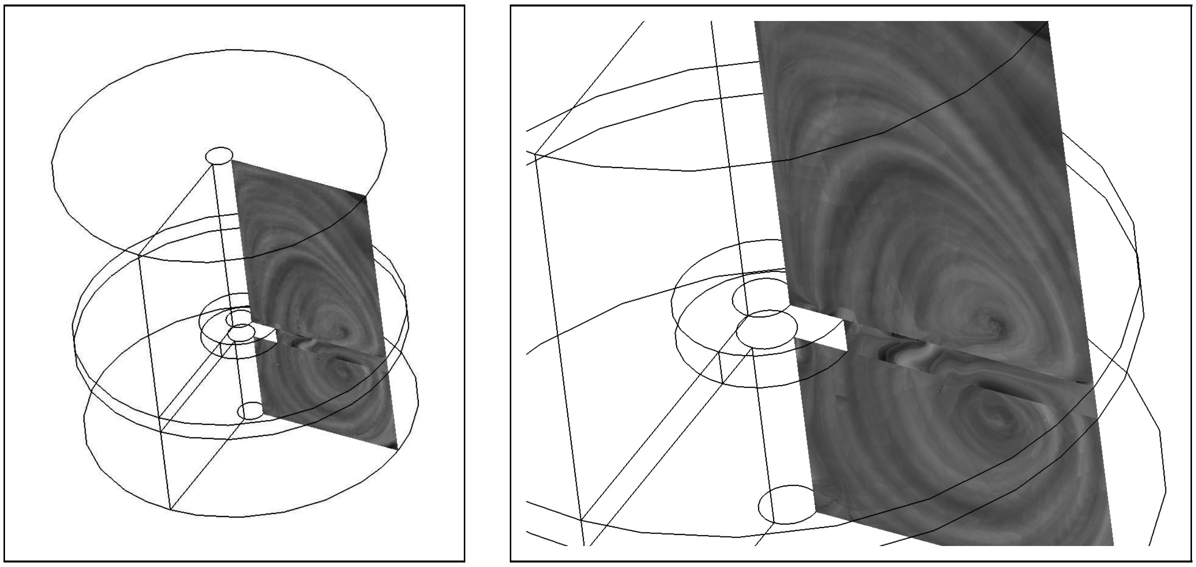

Figure 8: Here the flow in a sewage purification plant is visualized. And the input intensity is received by interpolating the randomly given intensities at the nodes. The aim of this application is to design a stirrer which mixes oxygen into the sewage in an optimal way. The multiblock data arises out of measurement.

6 Acknowledgements

We are grateful to Gunther Brenner from the Lehrstuhl für Strömungsmechanik of the University of Erlangen and to Dynamit Nobel AG for making the data set of an air bag available to us. Also we would like to thank Jakob Fainberg from the Lehrstuhl für Werkstoffwissenschaften VI of the University of Erlangen for a data set of a Czochralski furnace for crystal growing. Finally, we thank Marcus Schäfer from the Lehrstuhl für Strömungsmechanik of the University of Erlangen for a stirrer data set.

References

[1] B. Cabral, L. Leedom: *Imaging Vector Fields Using Line Integral Convolution.* SIGGRAPH, Computer Graphics Proceedings, 263–270, 1993.

[2] R. W. Clough, J. L. Tocher: *Finite element stiffness matrices for the analysis of plate bending.* Proceedings of 1st Conference on Matrix Methods in Structural Mechanics, Wright-Patterson, 515–545, 1965.

[3] P. Deuflhard, F. Bornemann: *Numerische Mathematik II, Integration gewöhnlicher Differentialgleichungen.* Walter de Gruyter Lehrbuch, Berlin, 1994.

[4] L. K. Forssell: *Visualizing Flow Over Curvilinear Grid Surfaces Using Line Integral Convolution.* IEEE Computer Society, Proceedings of Visualization, 240–247, 1994

[5] R. Grosso, M. Schulz, J. Kraheberger, T. Ertl: *Flow Visualization for Multiblock Multigrid Simulations.* Proceedings of 7th Eurographics Workshop on Visualization in Scientific Computing, Prague, P. Slavick, J. van Wijk (eds), Springer-Verlag, 1996.

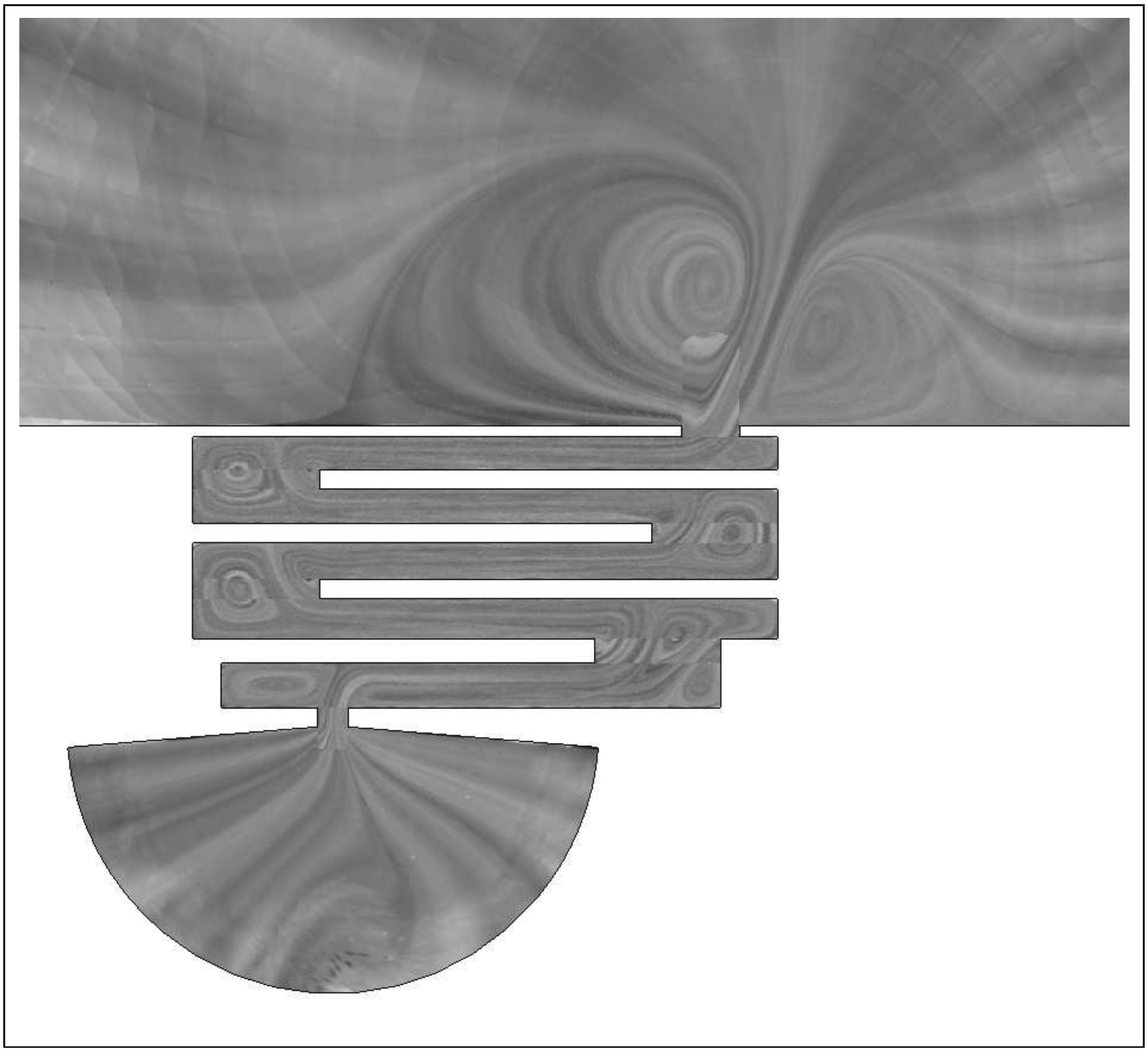

Figure 9: Slice through the gas generator of a car air bag. The velocity field of the unsteady gas flow is visualized during the initial ignition of the air bag. At the bottom you can see the box with the blasting composition, then a channel to reduce the temperature of the gas and on the top a small part of the actual air bag. Here the interpolation method has been used to compute the input intensities. This data set consists of thirteen blocks and you can detect block boundaries because one point on a block boundary generally has two different intensities in the two different blocks where it belongs to.

[6] R. Haimerl: *Textur-basierte Verfahren in der Strömungsvisualisierung.* Diplomarbeit, Erlangen, 1996.

[7] J. L. Helman, L. Hesselink: *Visualizing vector field topology in fluid flows.* IEEE Computer Graphics and Applications, 36–46, 1991.

[8] J. Hoschek, D. Lasser: *Grundlagen der geometrischen Datenverarbeitung.* Teubner-Verlag, Stuttgart, 1992.

[9] J. P. M. Hultquist: *Interactive numerical flow visualization using stream surfaces.* Computing Systems in Engineering, 349–353, 1990.

[10] W. C. de Leeuw, J. J. van Wijk: *A Probe for Local Flow Field Visualization.* IEEE Computer Society, Proceedings of Visualization, 39–45, 1993.

[11] N. Max, B. Becker, R. Crawfis: *Flow volumes for interactive vector field visualization.* IEEE Computer Society, Proceedings of Visualization, 19–24, 1993.

[12] M. J. D. Powell, M. A. Sabin: *Piecewise quadratic approximations on triangles.* ACM Transactions on Mathematical Software 3, 316–325, 1977.

[13] D. Stalling, H.-C. Hege: *Fast and Resolution Independent Line Integral Convolution.* SIGGRAPH, Computer Graphics Proceedings, 249–256, 1995.

[14] J. J. van Wijk: *Spot noise-texture synthesis for data visualization.* SIGGRAPH, Computer Graphics Proceedings, 309–318, 1991.