Interner Bericht

Streamball Techniques for Flow Visualization

Manfred Brill, Hans Hagen, Hans-Christian Rodrian
Computer Science Department, University of Kaiserslautern
Germany

Wladimir Djatschin, Stanislav V. Klimenko
Institute for High Energy Physics (IHEP)
Protvino, Russia

244/94

Fachbereich Informatik

Universität Kaiserslautern · Postfach 3049 · D-67653 Kaiserslautern

Streamball Techniques for Flow Visualization

Manfred Brill, Hans Hagen, Hans-Christian Rodrian
Computer Science Department, University of Kaiserslautern
Germany

Wladimir Djatschin[*], Stanislav V. Klimenko
Institute for High Energy Physics (IHEP)
Protvino
Russia

April 1994

Abstract

We introduce the concept of streamballs for fluid flow visualization. Streamballs are based upon implicit surface generation techniques adopted from the well-known metaballs. Their property to split or merge automatically in areas of significant divergence or convergence makes them an ideal tool for the visualization of arbitrary complex flow fields. Using convolution surfaces generated by continuous skeletons for streamball construction offers the possibility to visualize even tensor fields.

1 Introduction

Streamlines, streaklines, pathlines and timelines play an important role in flow visualization. Most of these terms are directly derived from experimental flow visualization, where the corresponding phenomena are generated by inserting foreign material into the flow and observing it while it moves through the field.

- A streakline is produced by continuously injecting material like smoke or little hydrogen bubbles into the flow at a certain point and watching the resulting cloud of particles.

[*]currently on sabbatical at the Computer Science Department in Kaiserslautern

- Pathlines can be obtained by putting small objects into the flow field and exposing a photograph for a longer time, thus depicting the traces of these objects over time.
- Timelines are given by observing a line of particles flowing with the stream and making snapshots at several time steps.
- Streamlines eventually are defined as curves tangent to the velocity field in every point.

For steady flows streaklines, pathlines and streamlines obviously coincide [5].

Though all of these constructions are powerful tools for the investigation of two-dimensional fields, they are not very well suited for the visualization of three-dimensional field data as they heavily suffer from display ambiguities when being displayed in 2D. By interpolating several streak-, path-, stream- or timelines they can be used to build time surfaces or streak-, path- and stream ribbons, -tubes or -surfaces, which in conjunction with standard lighting and shading techniques often provide a much better impression of the overall topology of a 3D flow field. Furthermore, local parameters of the field can be mapped onto these surfaces and thus be displayed together with the field's velocity structure.

Many new techniques for flow visualization have been presented in the last few years [5].

Schroeder [6] introduced the stream polygon technique, where an n-sided, regular polygon perpendicular to the local velocity vector can be traced along a streamline to produce a three-dimensional sweep. Effects like twist or scalar parameters of the field can easily be displayed by accordingly rotating and shearing the polygon or changing attributes like radius or color while the polygon is swept through the field.

Another well-known method is the generation of stream surfaces by connecting adjacent streamlines with polygons.

Special care has to be taken whenever divergence of the flow causes adjacent streamlines to separate or convergence causes streamlines to coincide, as the polygonal approximation may become poor in these cases.

Hesselink and Helman [3] proposed an algorithm which connects the critical points of the vector field on the surface of an object to a two-dimensional skeleton. This skeleton represents the global topology of the flow on this surface. Starting from points on the skeleton, streamlines in the flow around the object are calculated. By tesselating adjacent streamlines, stream surfaces are built. To avoid splitting of the stream surfaces in areas of divergence, the surfaces are recursively refined by introducing additional starting points for the streamline calculations.

Hultquist [4] introduced an algorithm which simultaneously traces a set of particles generated by discretizing an originating curve through the field and connects the resulting paths with triangles. In this way, an advancing front of a steadily growing stream surface is obtained. Whenever the divergence between two of these particles becomes too big, new particles are inserted into the front; when particles come too close to each other, some of them are removed.

Van Wijk proposed the usage of so-called surface particles [7] for flow visualization. With this technique, a big number of particles released from a number of particle sources are traced through the flow field. By positioning the particle sources on a curve and displaying all particles as small geometric primitives shaped and coloured in dependency of certain field parameters the impression of stream surfaces textured according to local parameters can be given.

Quite a different approach which guarantees the generation of smooth stream surfaces also has been introduced by Van Wijk [8]. The central concept of Van Wijk's method is the representation of stream surfaces as implicit surfaces $f(x) = C$ representing the sweep of an initial curve through the field. The shape of the initial curve is defined by the value of f at the boundary of the region of interest. To calculate f, Van Wijk proposes two methods: solving the convection equation or tracing backwards the trajectories of grid points. The same technique can be used for the construction of time surfaces or stream volumes.

The purpose of this article is to present a new method for flow visualization based upon implicit surface generation techniques adopted from metaballs. We call the resulting objects streamballs.

In particular, streamballs are distinguished by their ability to split automatically or merge with each other depending on their distances. By advancing appropriate skeletons through the field and displaying the resulting streamballs, streak-, stream-, path- and timelines as well as -surfaces or -volumes can easily be visualized, no matter how complex the field to consider may be.

First, we introduce the concept of streamballs defined by a set of discrete centerpoints and their usage in flow visualization. Some examples are presented to show the benefits of this approach. To open up a wider range of visualization possibilities we then introduce the concept of streamballs built of continuous two-dimensional skeletons. After presenting some examples we finally give a short summary and concluding remarks.

2 Streamballs

2.1 Streamballs with discrete skeleton

2.1.1 Basic concept

In 1982, Blinn [1] introduced the usage of implicit surfaces to display molecular compounds. He generated a potential field F defined by a finite set S of centerpoints s_i, which at a given point x in space is given as the sum of weighted influence functions $I_i(x)$ generated by each of these centers:

$$F(S,x) = \sum_i w_i I_i(x) = \sum_i w_i e^{-a_i f_i(x)}, \tag{1}$$

$f_i(x)$ describing the shape, a_i the size and w_i the strength of the potential field.

Based on this field an isosurface $F(S,x) = C$ is constructed.

For example, if there is only one centerpoint s_1 and if $a_1 = \frac{1}{R_1^2}$ and $f_1(x) = ||x - s_1||^2$, the resulting isosurface will be a sphere with a radius proportional to R_1.

G.Wyvill et. al. [9] used a similar technique to construct what they called soft objects. To avoid both the computation of the exponential function and the square root calculations when computing the distance between x and a centerpoint, they applied the following polynomial approximation:

$$I_i(x) = \begin{cases} a\frac{r_i(x)^6}{R^6} + b\frac{r_i(x)^4}{R^4} + c\frac{r_i(x)^2}{R^2} + 1 & : \; r_i(x) \leq R \\ 0 & : \; r_i(x) > R \end{cases} \tag{2}$$

with $r_i(x) = ||x - s_i||$ and a, b, c chosen to satisfy

$$\begin{array}{ccc} I_i(0) = 1 & & I_i(1) = 0 \\ I_i'(0) = 0 & & I_i'(1) = 0 \\ & I_i(0.5) = 0.5 & \end{array} \tag{3}$$

The described primitives are commonly known as metaballs. Metaballs are distinguished by numerous useful properties:

- a single centerpoint generates a single, spherical surface
- as two centerpoints come close, their corresponding shells blend smoothly
- if two or more centerpoints coincide, a single, larger sphere is produced (In fact, if the value of C is chosen properly, the sphere generated by two of such centers will have exactly twice the volume as a sphere produced by one single centerpoint)
- as two centerpoints separate, the blending process is reversed
- CSG operations can be applied.

2.1.2 Discrete streamballs

The basic idea for visualizing flow data with streamballs is to release a couple of particles into the flow and to use their positions at several time steps as a sceleton for metaballs, which then by blending with each other form streamlines, stream surfaces etc. Referring as well to the term metaballs as to the usage of discrete skeletons we like to call all these objects discrete streamballs.

To represent a streamline using discrete streamballs we simply distribute a couple of centerpoints along this streamline close enough to each other to let them blend (Fig. 1). In this way a continuous and smooth "stream spaghetti" will be produced.

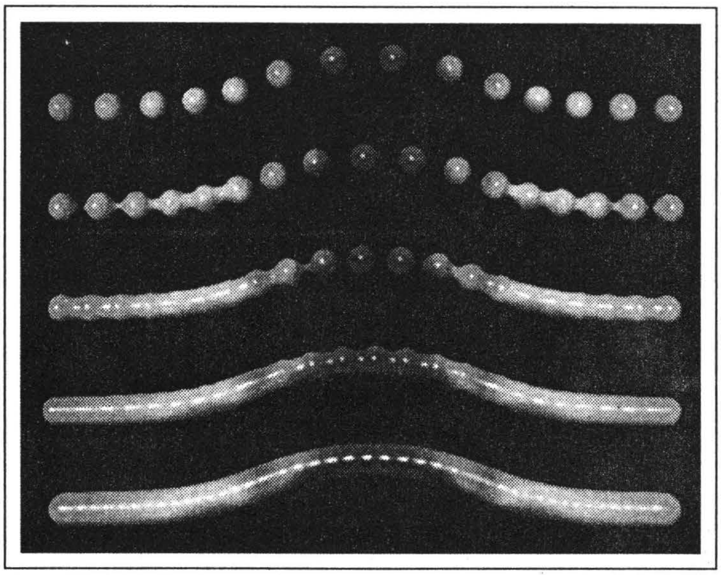

Figure 1: *Here the blending process of the streamballs is shown. By using an increasing number of centerpoints a three-dimensional streamline is produced.*

Stream surfaces are simply built by a set of such "spaghettis" originating on a single curve and close enough to each other to let the resulting blend produce a continuous smooth surface (Fig. 2).

Time surfaces for any given time $t = t_0 + \Delta t$ are built by distributing centerpoints on a surface at a time $t = t_0$ (this "seed" time surface can be found far from obstacles where the flow is parallel) and letting them flow with the field for a time Δt (Fig. 3).

Stream volumes of arbitrary initial shape are generated by advancing a cloud of particles through the flow field, which are ordered in respect to the desired shape.

Streamballs have the convenient property to split automatically in areas of significant divergence and to merge with each other in areas where convergence occurs. Thus, streamballs

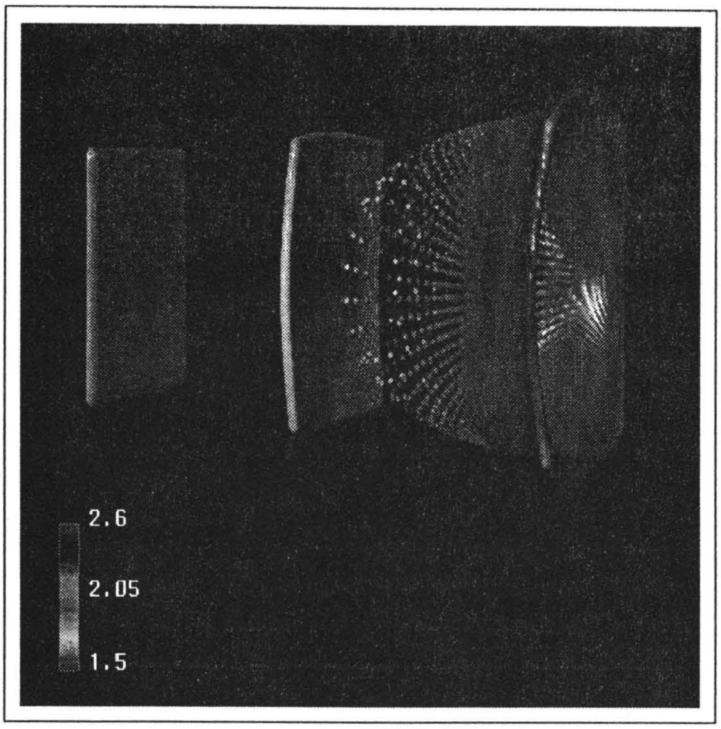

Figure 2: *Here the flow around an obstacle produced by the combination of a source and a sink is shown. Watch how the streamballs split on the obstacle and merge again behind it. Color is used to map the velocity of the flow.*

will not necessarily produce closed stream surfaces. This behaviour is an obvious consequence of the properties of metaballs. The way in which streamballs are behaving in such cases, however, can give valuable information on the structure of the flow. To produce closed surfaces nevertheless, simple oversampling techniques can be applied.

2.2 Streamballs with continuous skeletons

2.2.1 Basic concept

Bloomenthal and Shoemake [2] generalized the idea of metaballs proposing the usage of an arbitrary skeleton consisting of a continuum of points (i.e. lines, curves etc.) instead of a limited number of centerpoints to generate the influence function.

The field function F is given by the convolution of the Skeleton's characteristic function $\chi(S)$ with the weighted influence function $I(x)$:

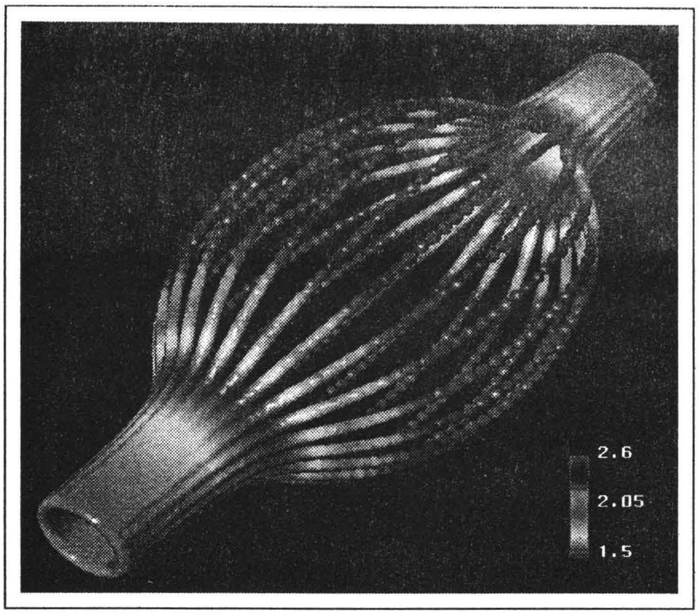

Figure 3: *This figure simultaneously shows three snapshots of a time surface encountered upon the same obstacle as in the previous figure. Though the time surface is splitting on the obstacle, the obstacle's shape can clearly be seen.*

$$F(S,x) = \chi(S) * \omega I(x) = ((\omega I) * S)(x) = \int\limits_{\xi \in S} e^{-\frac{||x-\xi||^2}{2}} d\xi \tag{4}$$

The convolution surface is given by building an isosurface $F(S,x) = C$.

To get reasonable computation times, we used an influence function similar to the one we already used for the streamballs with discrete skeleton:

$$I(x,\xi) = \begin{cases} af(x,\xi)^3 + bf(x,\xi)^2 + cf(x,\xi) + 1 & : \quad f(x,\xi) \leq 1 \\ 0 & : \quad f(x,\xi) > 1 \end{cases} \tag{5}$$

and

$$f(x,\xi) = \frac{||x-\xi||^2}{R^2} \tag{6}$$

again with a, b and c chosen to satisfy the conditions (3).

The objects generated this way preserve all useful properties of the implicit surfaces as they were introduced by Blinn.

2.2.2 Continuous streamballs

Convolution surfaces with continuous skeletons are a powerful tool for flow visualization, as they provide the ability to produce perfectly smooth surfaces around their skeletons. Instead of using a set of points advancing through the field as the skeleton for an implicit surface, we just use these points to construct an adequate continuous skeleton which in turn generates the implicit surface.

To represent a stream line, we trace a point along this streamline in several discrete time steps and connect the single point positions to build the skeleton of the streamball. The resulting three-dimensional streamline generally will be much thinner and smoother than one produced by streamballs with a discrete skeleton.

Stream surfaces are simply built of a set of 3D streamlines very close to each other (Fig. 4).

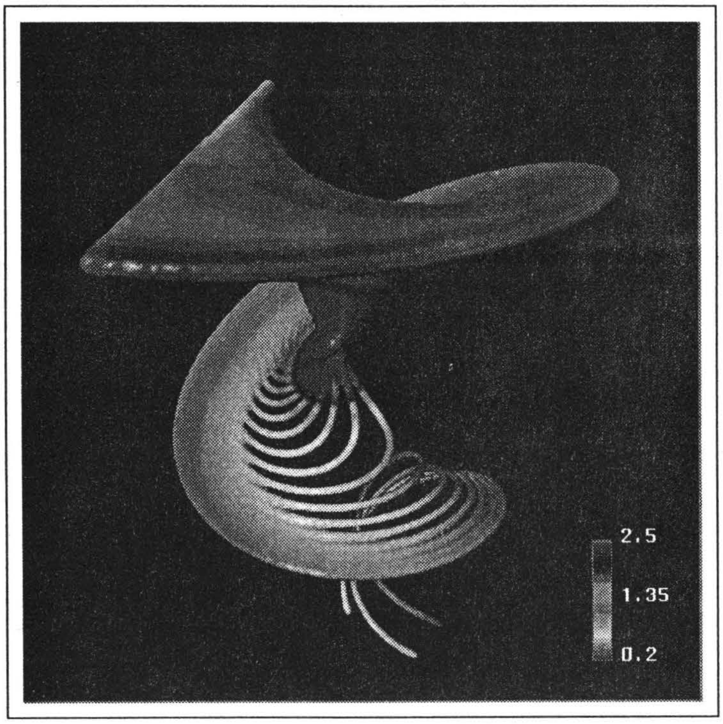

Figure 4: *Here a stream surface produced by a rake of 100 3D-streamlines in a field containing a vortex is shown.*

Similar, time surfaces are built of a couple of three-dimensional timelines.

When time surfaces are tracked through the field, special attention has to be paid on obstacles to prevent the time surfaces from being pulled "through" the obstacle. This problem can be overcome by controlling the length of the single skeleton segments and dividing them if necessary.

2.3 Mapping of local field parameters

Streamballs imply a variety of possibilities for the mapping of local field parameters.

With discrete streamballs, for example, an easy method to map the course of a local parameter along a streamline would be to change the radius of the single primitives according to the parameter's value at the considering centerpoints. The result will be a three-dimensional streamline with diameter corresponding to the magnitude of the parameter to map (Fig. 5). By adding color or applying standard texturing and mapping techniques many scalar values may be displayed simultaneously.

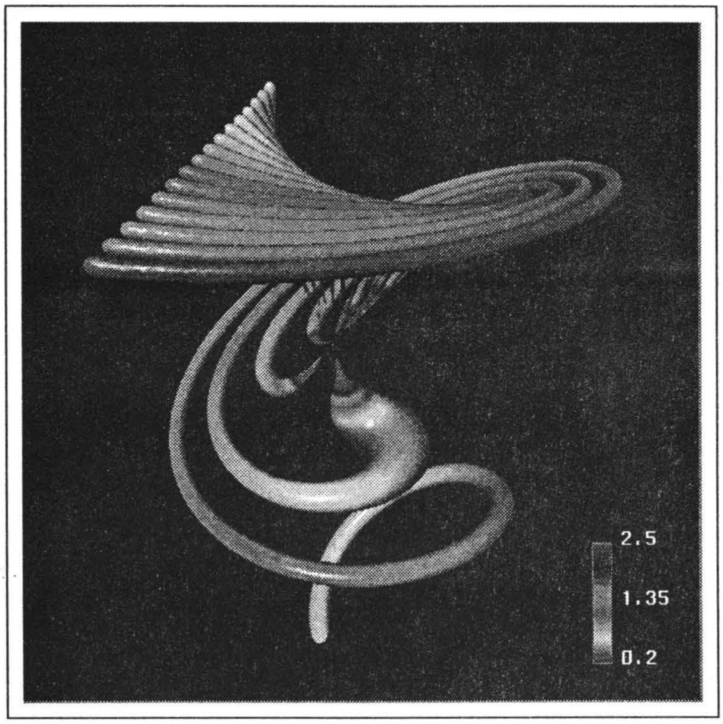

Figure 5: *This is a picture of a couple of discrete streamballs in the same field as in the previous picture. We used both the radius as well as the color of the streamballs to show velocity.*

Using continuous streamballs, it is possible to construct a local coordinate system (ξ_1, ξ_2, ξ_3) oriented in dependency of the field's structure in every point of the skeleton. This coordinate system can be used to build a non-axisymmetrical influence function. The resulting surface will have a non-axisymmetrical cross-section with orientation and diameter corresponding to the structure of the field.

Similar to a technique introduced by Hesselink and Helman [3], this property can be exploited for the representation of tensor data. For this purpose the skeleton is directed along a so-called hyperstreamline (a curve tangent to the main eigenvector of a tensor field) with ξ_2 and ξ_3 oriented parallel to the two eigenvectors which are perpendicular to the main eigenvector, respectively. Using some standard mapping technique to show the absolute value of the main eigenvector and choosing the length of the two axes of the ellipse corresponding to the absolute values of the remaining two eigenvectors, we can represent not only direction, but even the magnitude of all three eigenvectors at the same time.

3 Summary

The proposed technique proves to be useful for 3-D flow visualization in several ways:

- construction of stream lines, stream surfaces and stream volumes as well as time surfaces is possible in a quite easy and natural way
- additional information on the flow is given by the way in which streamballs divide or blend in areas of significant divergence or convergence
- the algorithm obviously serves well even in cases of very complex flow fields
- streamballs provide powerful mapping possibilities for flow-related parameters and so are not only applicable for the examination of vector fields, but even can be used for the exploration of the complex structure of tensor fields.

Acknowledgements

The research for this project is funded by a grant of the "Stiftung Innovation für Rheinland-Pfalz" awarded to the University of Kaiserslautern. Wladimir Djatschin is supported by the DAAD.

Many thanks to Henrik Weimer for programming and helpful discussions.

References

[1] J. Blinn: *A Generalization of Algebraic Surface Drawing*, ACM Transactions on Graphics 1, pp. 235-256

[2] J. Bloomenthal, K. Shoemake: *Convolution Surfaces*, Computer Graphics 25(4) 1991, pp. 251-256

[3] J. L. Helman, L. Hesselink: *Visualizing Vector Field Topology in Fluid Flows*, IEEE Computer Graphics & Applications 91, pp. 36-46

[4] J.P.M. Hultquist: *Constructing Stream Surfaces in Steady 3D Vector Fields*, Proceedings on Visualization 92, pp. 171-178

[5] F. H. Post, Theo. van Walsum: *Fluid Flow Visualization*, Focus on Scientific Visualization, Springer, 1993, pp. 1-40

[6] W. Schroeder, C. Volpe, W. Lorensen: *The Stream Polygon: A Technique for 3D Vector Field Visualization*, Proceedings on Visualization 91, pp. 126-132

[7] J.J. van Wijk: *Rendering Surface Particles*, Proceedings on Visualization 92, pp. 54-61

[8] J.J. van Wijk: *Implicit Stream Surfaces*, Proceedings on Visualization 93, pp. 245-252

[9] G. Wyvill, C. McPheeters, B. Wyvill: *Data structure for soft objects*, The Visual Computer 1986(2), pp. 227-234