Streamball Techniques for Flow Visualization

Manfred Brill
Hans Hagen
Hans-Christian Rodrian

Computer Science Department
University of Kaiserslautern
Germany

Wladimir Djatschin
Stanislav V. Klimenko

Computer Science Department
Institute for High Energy Physics (IHEP)
Russia

Abstract

We introduce the concept of streamballs for 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

1.1 Streamlines and stream surfaces

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.

- Streaklines are produced by continuously injecting material like smoke or little hydrogen bubbles into the flow at certain points and watching the resulting clouds of particles.
- 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 finally are defined as curves tangent to the velocity field in every point.

For steady flows streaklines, pathlines and streamlines obviously coincide [7].

In computational flow visualization, streak-, path-, time-, and streamlines are often simulated to get an insight into the structure of a flow field. Though 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 two dimensions on a computer screen. Therefore, streak-, path-, stream-, or timelines are often used to build time surfaces or streak-, path-, and stream ribbons, -tubes, or -surfaces, which in conjunction with standard lighting and shading techniques can provide a much better idea 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.

1.2 Previous work

Many new techniques for flow visualization have been presented in the last few years [7].

Schroeder et. al. [8] introduced the stream polygon technique, where n-sided, regular polygons perpendicular to the local velocity vector are placed along a streamline. Effects like twist or scalar parameters of the field are displayed by accordingly rotating and shearing the polygons or changing attributes like radius or color. By sweeping stream polygons along streamlines, three-dimensional stream tubes can be built.

1070-2385/94 $04.00 © 1994 IEEE

Another 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 come very close to each other, as the polygonal approximation may become poor in these cases.

Helman and Hesselink [4] proposed an algorithm which connects the critical points of the vector field on the surface of an object to form a two-dimensional skeleton. This skeleton represents the global topology of the flow on the 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 [5] introduced an algorithm which simultaneously traces a set of particles originating from discrete points on some 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 [9] proposed the usage of so-called surface particles 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.

A different approach which guarantees the generation of smooth stream surfaces was also introduced by Van Wijk [10]. The central concept of this 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 inflow boundaries. 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.

1.3 Overview

In this article we 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 or merge with each other automatically 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 given field may be. The mathematical representation of streamballs offers a variety of mapping possibilities for parameters of the flow field.

Section 2.1 introduces the concept of streamballs defined by a set of discrete centerpoints and their usage in flow visualization. In Section 2.2, the concept of streamballs constructed from continuous two-dimensional skeletons, which open up a wider range of visualization possibilities, is introduced. In Section 2.3, some mapping techniques for streamballs are presented. The rendering method that we used is described in Section 2.4. Finally, Section 3 contains a short summary and concluding remarks.

2 Streamballs

2.1 Streamballs with discrete skeletons

2.1.1 Basic concept

In 1982, Blinn [1] introduced the usage of implicit surfaces to display molecular compounds. With his method, a potential field F defined by a finite set S of centerpoints s_i is used to generate an implicit surface which represents the molecules.

At a given point x in space, $F(S,x)$ is given as the sum of weighted influence functions $I_i(x)$ generated by each of these centers:

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

where $f_i(x)$ describes 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 $a_1 = \frac{1}{R^2}$ and $f_1(x) = ||x - s_1||^2$, the resulting isosurface will be a sphere whose radius depends on R.

G. Wyvill et. al. [11] used a similar technique to construct what they called soft objects. To localize

the influence of the centerpoints and to avoid the computation of the exponential function, they applied the following polynomial approximation:

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

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

$$\begin{aligned} &I_i(0) = 1 \quad I_i(0.5) = 0.5 \quad I_i(R) = 0 \\ &I_i'(0) = 0 \qquad\qquad\qquad\quad I_i'(R) = 0 \end{aligned} \tag{3}$$

The described primitives are commonly known as metaballs or blobby objects. 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, i.e. the resulting surface is C^∞-continuous.
- 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.

2.1.2 Discrete streamballs

The basic idea for visualizing flow data with streamballs is to use the positions of particles in the flow as centerpoints for implicit surfaces, which then by blending with each other form three-dimensional streamlines, stream surfaces etc. The centerpoints can be looked upon as a discrete skeleton of the surface constructed in this way. Referring both to the term metaballs and to the usage of discrete skeletons we call the resulting three-dimensional objects discrete streamballs.

To represent a streamline using discrete streamballs we simply distribute a number of centerpoints along this streamline close enough to each other to let the surrounding isosurfaces blend. This blending process is shown in Figure 1. By increasing the number of centerpoints s_i step by step, a continuous, three-dimensional representation of a streamline is produced.

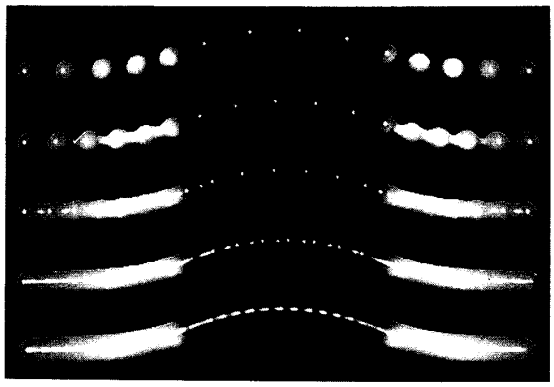

Figure 1: *The blending process of the streamballs.*

To construct stream surfaces, a number of particles originating from different positions on some starting curve are advanced through the flow field. Their positions at several time steps are used as a skeleton for the streamballs. When the particles initially are close to each other, they will produce a continuous and smooth surface, which will split automatically in areas where divergence occurs, and merge automatically in areas of convergence. An example for this can be seen in Figure 2, where the flow around an obstacle, simulated by the combination of a source and a sink, is shown. Notice how the streamballs split around the obstacle and merge again behind it. Color is used to map the velocity of the flow.

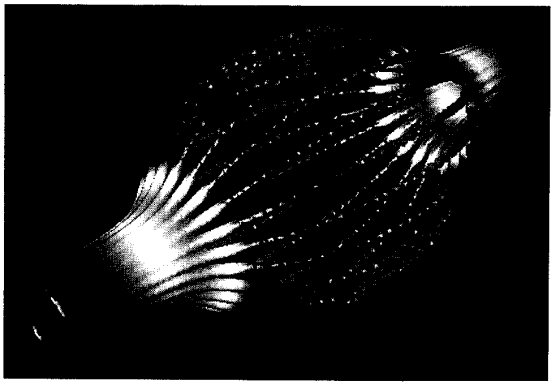

Figure 2: *Discrete streamballs flowing around an obstacle.*

Time surfaces for any given time $t = t_0 + \Delta t$ are built by distributing skeleton points on a starting surface at a time $t = t_0$ and letting them flow with the

(See color plates, page CP-25.)

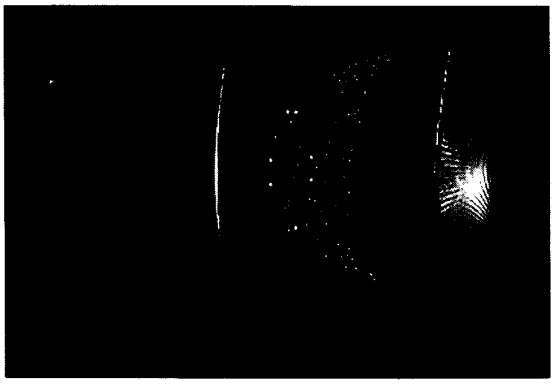

Figure 3: *Three snapshots of a time surface hitting an obstacle.*

field for a time Δt. Figure 3 simultaneously shows three snapshots of a time surface hitting the same obstacle as in Figure 2. Though the time surface splits on the obstacle, the obstacle's shape can clearly be seen.

Stream volumes of arbitrary initial shape are generated by advancing a cloud of particles through the flow field, which are initially arranged to form the desired shape, and using their positions over time as a skeleton for the streamballs.

As can be seen from the figures, streamballs have the convenient property to split automatically in areas of significant divergence and to merge with each other in areas where convergence occurs. This behavior is a natural consequence of the properties of metaballs.

Thus, streamballs will not necessarily produce closed stream surfaces. The way in which streamballs behave in such cases, however, can give valuable information on the structure of the flow. In order to produce closed surfaces nevertheless, one can simply release additional particles in areas of high divergence.

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 $\chi_S(x)$ with the weighted influence function $I(x)$:

$$F(S,x) = \chi_S(x) * \omega I(x) = \int_{\xi \in S} \omega(\xi) I(\xi, x) d\xi \qquad (4)$$

Using an exponential influence function, we get

$$F(S,x) = \int_{\xi \in S} \omega(\xi) e^{-\frac{||x-\xi||^2}{2}} d\xi \qquad (5)$$

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 skeletons:

$$I(\xi,x) = \begin{cases} af(\xi,x)^3 + bf(\xi,x)^2 + \\ \qquad + cf(\xi,x) + 1 : f(\xi,x) \leq 1 \\ \qquad\qquad 0 \quad : f(\xi,x) > 1 \end{cases} \qquad (6)$$

and

$$f(\xi,x) = \frac{||x-\xi||^2}{R^2} \qquad (7)$$

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

The objects generated in this way preserve all useful properties of Blinn's implicit surfaces.

2.2.2 Continuous streamballs

Convolution surfaces with continuous skeletons are a powerful tool for flow visualization. They provide the ability to produce perfectly smooth surfaces around their skeletons.

With the discrete streamballs, we used a set of particle positions as a skeleton for an implicit surface. Now we use these points to construct a continuous skeleton which in turn generates the implicit surface.

To represent a streamline, for example, we trace a particle along this streamline through several discrete time steps and connect the single particle positions to build the skeleton of the streamball. The resulting three-dimensional streamline generally will be thinner and more regular than one produced by discrete streamballs using the same points as a skeleton.

Stream surfaces again are constructed from a set of three-dimensional streamlines which are very close to each other (Figure 4).

Similarly, time surfaces are built of a number of three-dimensional timelines.

(See color plates, page CP-25.)

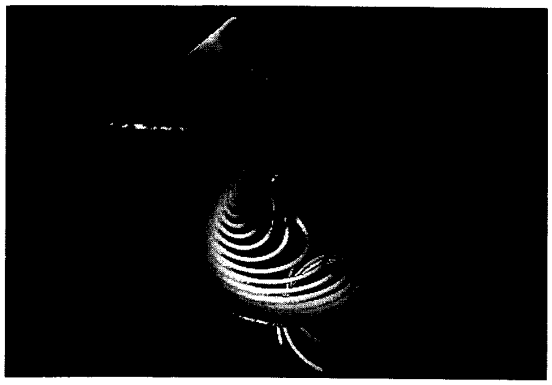

Figure 4: *A stream surface produced by a rake of 100 3D-streamlines in a flow field containing a vortex.*

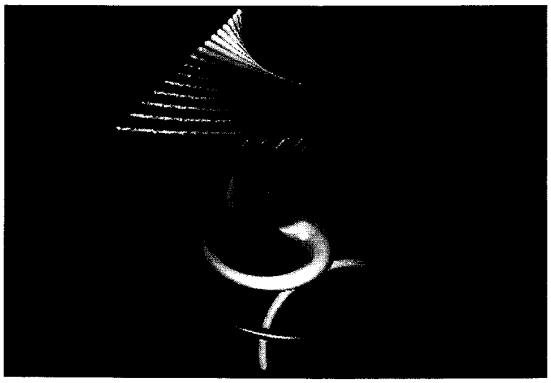

Figure 5: *Discrete streamballs in a flow field containing a vortex. Both radius and color show velocity.*

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

2.3 Mapping of local field parameters

Streamballs offer a variety of possibilities for the mapping of local field parameters. Besides standard mapping techniques, new mapping techniques based on the mathematical representation of the streamballs can be applied.

With discrete streamballs, for example, an easy method to map the value of a local parameter along a streamline is to choose the radius of the influence functions of each skeleton point according to the parameter's value at that point. The result is a three-dimensional streamline whose diameter corresponds to the magnitude of the parameter to map. In Figure 5 we used this technique to map the velocity of the flow.

A similar method has been used for the upper layer of the streamballs shown in Figure 6. The radius of the influence function of these streamballs was increased at several discrete positions along the streamline. By choosing the distances of these positions dependent on the absolute value of the velocity of the field, a good idea of this parameter's value along the streamline is given. It can be seen clearly that velocity is lower in front of the obstacle and higher on the side of it. To increase the radius of the influence functions at certain positions, we just placed discrete streamballs, each with a skeleton consisting of exactly one centerpoint, along the streamline.

A different method has been used to map the velocity on the surface of the lower layer of streamballs in Figure 6.

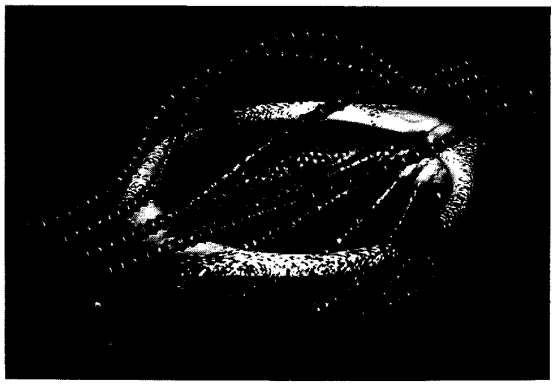

Figure 6: *Different mapping methods using streamballs.*

With this mapping technique local scalar parameters are mapped as roughness of the surface. For this, the amplitude of a three-dimensional oscillating function $F(x)$ is modulated by the value of velocity. The potential function of the streamballs then is superimposed by $F(x)$. For simplicity, we choose F to be

$$F(x) = sin(f_1 x_1) sin(f_2 x_2) sin(f_3 x_3), \qquad (8)$$

where the f_i are (not necessarily different) frequencies.

As the described mapping techniques influence only the geometry of our streamballs, more common mapping techniques, using e.g. material properties of the

(See color plates, page CP-25.)

surface, can be applied simultaneously. In all our figures we additionally used simple color mapping to depict velocity.

The streamballs in the middle of Figure 6 show a different color mapping method. The component of the velocity which describes the deviation of the flow from the central axis is mapped as color spots on the surface of these streamballs. The density of the color spots depends on the absolute value of the considered velocity component.

Similar to a technique introduced by Hesselink and Delmarcelle [3], continuous streamballs can be used 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). At every point of the skeleton a local coordinate system (ξ_1, ξ_2, ξ_3) is used with ξ_1 tangent to the main eigenvector e_1 at this point and ξ_2 and ξ_3 oriented in the directions of the two eigenvectors e_2 and e_3 which are perpendicular to the main eigenvector. Choosing the radius of the influence function in the directions of ξ_2 and ξ_3 corresponding to the absolute values of the two eigenvectors e_2 and e_3, an asymmetrical influence function can be constructed. The resulting isosurface will have an asymmetrical cross section with orientation and diameter dependent on the directions and eigenvalues of these two eigenvectors. Using some mapping technique to show the eigenvalue of the main eigenvector, it is possible to represent not only direction, but even the magnitude of all three eigenvectors at the same time.

2.4 Rendering

For rendering, the field function $F(S, x)$ of the streamballs is evaluated on a regular grid. The isosurface $F(S, x) = C$ is extracted from this grid by a simplified marching cubes algorithm, and the resulting triangles are rendered using Iris Explorer.

The modified marching cubes algorithm processes the grid slice by slice, so only two slices have to be held in memory at one time [6]. As the polynomial influence functions (2) or (6) are used, each part of the skeleton has only a local influence on the field function. Therefore, when computing the values of the field function for a grid point x, $F(S, x)$ has to be evaluated only for those parts of the skeleton which are close enough to that grid point to influence the potential field at x. This greatly reduces computation costs.

For high grid resolutions, as they may be necessary to see fine details, the huge number of triangles generated by the marching cubes algorithm is a considerable drawback. For this reason we are working on a fast adaptive triangulation algorithm, which will both reduce the number of field function evaluations and the number of triangles produced.

3 Summary

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

- The representation of streamlines, stream surfaces and stream volumes as well as time surfaces is possible in a quite easy and natural way.
- Streamballs split or merge automatically in areas of significant divergence or convergence. Valuable information on the flow is given by the way in which the streamballs divide or blend in these cases.
- Due to the underlying mathematical representation, streamballs provide powerful mapping possibilities for flow-related parameters. Hence, they are not only suited for the examination of vector fields, but can even be used for the exploration of the complex structure of tensor fields.
- Streamballs can be applied even in cases of very complex flow 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 discussion.

References

[1] J. Blinn: *A Generalization of Algebraic Surface Drawing*, ACM Transactions on Graphics Vol. 1, No. 3, 1982, pp. 235-256

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

[3] T. Delmarcelle, L. Hesselink: *Visualization of Second Order Tensor Fields and Matrix Data*, Proceedings of Visualization '92, pp. 316-323

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

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

[6] M. Matzat, R. H. van Lengen: *Marching-Cube-Algorithmus zur Oberflächenrekonstruktion medizinischer Daten*, Projektarbeit, Fachb. Informatik, Universität Kaiserslautern, 1994

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

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

[9] J. J. van Wijk: *Rendering Surface Particles*, Proceedings of Visualization '92, pp. 54-61

[10] J. J. van Wijk: *Implicit Stream Surfaces*, Proceedings of Visualization '93, pp. 245-252

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