Automatic Stream Surface Seeding

Matt Edmunds[1], Tony McLoughlin[1], Robert S. Laramee[1], Guoning Chen[2], Eugene Zhang[3], and Nelson Max[4]

[1]Visual and Interactive Computing Group
Department of Computer Science,
Swansea University, United Kingdom
{csmatti, cstony, r.s.laramee}@swansea.ac.uk

[2]Scientific Computing and Imaging (SCI) Institute
University of Utah, United States
chengu@sci.utah.edu

[3]School of Electrical Engineering and Computer Science,
Oregon State University
Corvallis, OR 97331, United States
zhange@eecs.oregonstate.edu

[4]Department of Computer Science,
University of California, Davis
Davis, California, United States
max@cs.ucdavis.edu

Abstract

The visualisation of 3D flow poses many challenges. Difficulties can stem from attempting to capture all flow features, the speed of computation, and spatial perception. Streamlines and stream surfaces are standard tools for visualising 3D flow. Although a variety of automatic seeding approaches have been proposed for streamlines, little work has been presented for stream surfaces. We present a novel automatic approach to the seeding of stream surfaces in 3D flow fields. We first describe defining seeding curves at the domain boundaries from isolines generated from a derived scalar field. We then detail the generation of stream surfaces integrated through the flow and discuss the associated challenges of surface termination and occlusion. We also present the results of this algorithm, how we achieve satisfactory domain coverage and capture the features of the flow field. Strategies for resolving occlusion resulting from seeding multiple surfaces are also presented and analysed.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Surface generation

1. Introduction and Motivation

Flow visualisation is a powerful means for exploring, analysing and communicating simulation or experimental results. The topic of flow visualisation using stream surfaces has become an increasingly important direction of research in recent years.

Streamlines are an intuitive, fast, and simple method for visualising flow. Streamlines require less computation than surfaces and are generally easier to implement than their surface counterparts. These types of curves can present disadvantages, however, such as visual clutter (when too many streamlines are rendered) and lack of depth perception. Surface primitives (as opposed to curves in space) have well defined normals. Thus they offer perceptual advantages including: lighting and shading which provide intuitive depth cues, the ability to texture map including texture advection [LGSH06], the placement of additional geometry on the surface [LMGP97], and their use for depicting boundaries. Surfaces generally suffer from less visual clutter than lines, points, or other geometric primitives because they offer greater spatial continuity. Stream surfaces partition the flow domain into regions of similar flow behaviour. The same cannot be said of stream lines in 3D flow fields.

Stream surfaces for visualisation face many challenges. These surfaces must represent an accurate approximation of the underlying simulation. Adequate sampling must be maintained while reducing the unnecessary computational overhead associated with over-sampling. When using surfaces the problem of occlusion arises. This may stem from multiple surfaces that occlude one another, a large surface that results in self occlusion, or a combination of both. A general solution to this problem is to use transparency. With integral surfaces we have additional options. Illustrative techniques can be used to improve perception. Also stream surface seeding positions may be modified to reduce clutter. Manual seeding is the most common method for the placement of stream surfaces. However interactive stream surface placement is based on trial and error. Important characteristics of the flow can easily be missed. Stream surfaces must be seeded such that they capture the features of the flow.

A significant body of research has been invested into automatic seeding strategies using streamlines, but, little has been offered for automatic stream surface seeding. This provides strong motivation for studying stream surfaces and their seeding.

The main benefits and contributions of this paper are:

- A novel automatic approach to seeding stream surfaces in 3D flow fields, including a new technique for termination of surfaces.
- Techniques for reducing occlusion related to seeding multiple surfaces.

We illustrate how to achieve adequate coverage of the domain and capture the features of the flow field. Our focus pays particular attention to the seeding curve generation, inter-surface awareness, and occlusion.

First a review of related literature is conducted in section 2. Then a detailed presentation of the algorithm is given in section 3. The results are discussed in section 4. The limitations of this paper are highlighted in section 5. Conclusions and proposed future work are mentioned in section 6.

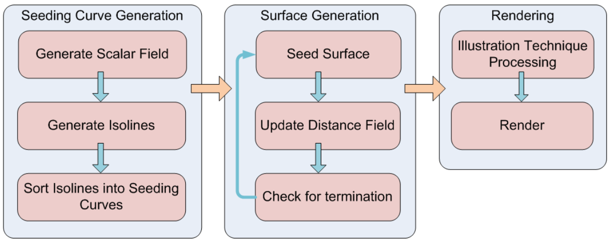

Figure 1: *Our algorithm pipeline. The first stage creates a scalar field on the domain boundary according to the angle of incidence of out flow. A set of isolines is then generated which are refined to create seeding curves for the streamsurfaces. The next stage is surface generation. The streamsurfaces are advanced using numerical integration through the velocity field. In addition to surface advancement a distance field is generated which describes the locality to existing surfaces. This allows us to limit the density of multiple surfaces. The final stage renders the surface using techniques such as colour coding and semi-transparency.*

2. Related Work

This section studies the different techniques and contributions related to seeding integral objects in 3D.

A streamline is a curve that is everywhere tangent to a vector field. A stream surface can be described as the union of all streamlines passing through a seeding curve. It can be approximated by generating a series of streamlines along a seeding curve and joining them to produce a polygonal representation. Stream surfaces are useful for understanding flow structures within a single time step or static flow field and are relatively simple to compute.

Zöckler et al. introduce a method of illuminating streamlines [ZSH96]. There is no native support for the lighting of line primitives in graphics libraries such as OpenGL, due to the fact that line primitives have no unique normal vector. A streamline placement algorithm has been introduced. For the placement technique a stochastic seeding algorithm is applied. The degree of interest in each cell is defined on some scalar value (i.e., velocity magnitude). See Weinkauf et al. [WT02] [WHN*03] for applications of this seeding strategy.

Mattausch et al. [MT*03] combine the illuminated streamlines technique of [ZSH96] with an extension of the evenly-spaced streamlines seeding strategy of Jobard and Lefer [JL97] to 3D.

Chen et al. [CCK07] present a novel method for the placement of streamlines that does not rely solely on density placement or feature extraction. This approach is based on a similarity method which compares candidate streamlines based on their shape and direction as well as their Euclidean distance from one another.

Li et al. [LS07] present a streamline placement strategy for 3D vector fields. This is the only approach of its kind where an image-based seeding strategy is used for 3D flow visualisation.

Interactive seeding strategies have been used in various modern, real-world applications including the investigation and visualisation of engine simulation data [Lar02] [LWSH04] [LGD*05].

An image-space-based method for placement of evenly-spaced streamlines on boundary surfaces is presented by Spencer et al. [SLCZ09]. The vector field is projected onto the image plane. Thus, the complexity of tracing in the large unstructured grids that typically result from CFD simulations is avoided. Streamline density is controlled by an adaption of the method of [JL97].

More recently Marchesin et al. [MCHM10] present a view-dependent strategy for seeding streamlines in 3D vector fields. No distribution of streamlines is ideal for all viewpoints. Therefore, this method produces a set of streamlines tailored to the current viewpoint.

While many papers have been published presenting detailed algorithms for the construction of stream surfaces, this is the first paper (to our knowledge) to focus specifically on a seeding algorithm for stream surfaces.

Space limitations prevent an overview of stream surface construction algorithms. For a complete overview of streamline seeding strategies, and integral surface construction algorithms see McLoughlin et al. [MLP*10].

3. Automatic Surface Seeding

This section presents the proposed automatic stream surface seeding algorithm, starting with an overview of the seeding pipeline illustrated in figure 1. This algorithm is described in three stages; seeding curve generation, surfaces computation, and rendering.

1. The starting point is the placement of seeding curves. This is realised by generating the seeding curves from isolines derived at the domain boundary. A scalar field is derived at the domain boundary based on the direction of flow exiting the domain. The isolines are constructed using a simple marching squares algorithm, so a simple point sorting technique is employed to store the vertices in the required order. Refer to section 3.1 for details.
2. Once the seeding curves are computed, the next step generates a collection of stream surfaces which advance through the vector field until they meet a predefined set of parameters. The termination parameters include maximum surface length and distance to neighbouring surfaces. The minimum distance between each surface is calculated using a distance field. Each surface is then terminated. When all surfaces are completed the algorithm proceeds to the next step. Section 3.2 provides details.
3. The final step in the pipeline is to render the scene. A number of techniques are implemented to reduce occlusion and aid the viewer in perceiving the resulting visualisation. This includes the use of transparency, colour, clipping planes, edge highlighting, lighting and shadow, and surface filtering. See section 3.3.

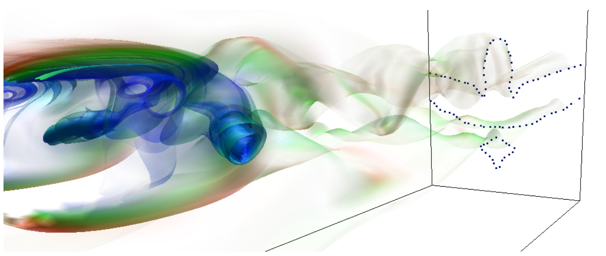

Figure 2: *A set of streamsurfaces on a simulation of flow behind a cuboid [CSBI05]. Transparency is mapped to surface curvature. The seeding curves can be seen on the far plane of the domain.*

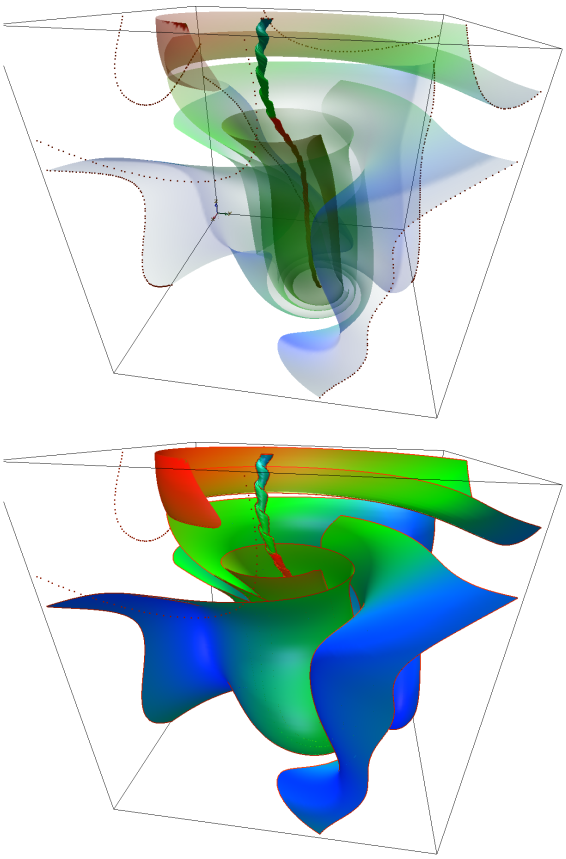

Figure 3: *A set of streamsurfaces seeded automatically using our technique on a tornado simulation. The top image shows surfaces with transparency mapped to surface curvature. This reduces occlusion and allows insight into the behaviour of the inner flow structures. The bottom image shows the same set of surfaces without transparency.*

3.1. Seeding Curve Generation

Generation of seeding curves from isolines derived at the domain boundary is performed in three steps. The first step is to define a scalar field based on exit flow at the domain boundary. The scalar represents the angle of incidence between the boundary vector and the domain at each sample. The calculation is performed by projecting the unit vector onto the domain boundary. The resultant magnitude is used as the scalar. If the exit trajectory is perpendicular to the domain boundary a scalar value of zero is stored, if the exit trajectory is parallel to the boundary then the scalar is stored as unity.

The next step is to construct the isolines from the scalar field at the boundary using a simple marching squares algorithm. The resulting vertices would normally be rendered as order independent line segments. However the vertices require correct ordering for the seeding curve.

3.2. Surface Generation

Stream surfaces are propagated from each of the seeding curves defined in the previous step [GKT*08]. The surfaces are then terminated according to a predefined set of parameters. Maximum surface length, boundary proximity, and distance to neighbouring surfaces are used to determine termination.

Calculating surface length and determining boundary proximity are straight forward. However a distance field is used for the efficient detection of neighbouring surfaces. As each surface is generated, its location is added to the distance field. Then the field is updated. As the next surface is propagated through the domain, it is tested against the distance field to determine if the proximity to any neighbouring surfaces is less than a predefined minimum distance. If so the surface propagation is terminated. This process is repeated for all surfaces.

After the initial set of stream surfaces is constructed from the domain boundaries, additional stream surfaces can be seeded from existing ones by a user-defined separating distance in order to gain complete domain coverage.

3.3. Rendering

Rendering of the scene is the final step in the pipeline. A number of techniques are implemented to aid the viewer's perception of the resulting visualisation and to aid in the reduction of occlusion. The techniques used to represent the results include the use of transparency, colour, silhouette edge highlighting, lighting and shadow, and surface filtering.

Colour is used to represent velocity magnitude, while opacity is mapped to surface curvature. Lighting and shading are standard tools to aid in the perception of depth, and shape. Silhouette edge highlighting is used to help the viewer in understanding where the surfaces curve away from the viewer, and enhance the perception of surface edges.

Another technique involves filtering of the surfaces to aid in the reduction of visual clutter. This is done by selecting which surfaces are to be displayed.

4. Results

A natural question to ask for this technique is whether we achieve satisfactory domain coverage and capture the features of the flow field. Figure 3 show results from seeding tornado data. It can be seen that the domain is adequately seeded to capture the structure of the tornado. Using translucency and silhouette edges improves the users perception of the results. This combined with filtering some of the generated surfaces aids in reducing the occlusion.

Modifying the initial isovalue used for the generation of the seeding curves produced results that although different, visualised the data equally well. A general isovalue of 0.5 was found to be adequate to produce good visualisations. The main point of using isolines derived from exit flow direction is the binding of the coherent flow structures at the boundary and tracing them through the domain.

Figure 2 demonstrates capturing the vortices generated behind the back face of the cuboid (cuboid not rendered for clarity). Used in conjunction with translucency the perception of the vortices are enhanced.

In figure 4 the seeding of the surfaces fills the domain, capturing the features of the flow. The complex flow structures are well represented with our technique. Some of the seeded surfaces are again filtered for improving the clarity of the rendering, this is further enhanced by using transparency to visualise important parts of the data otherwise occluded.

The illustrative strategies implemented for resolving occlusion resulting from seeding multiple surfaces improve perception and therefore aid understanding of the underlying flow

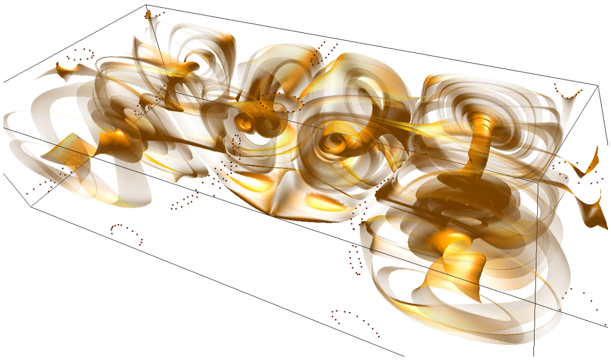

Figure 4: *The set of streamsurfaces using our method on Bernard flow. The use of transparency allows us to see the vortex structures that may otherwise be occluded due to self-occlusion and/or occlusion from other streamsurfaces.*

structures. Visualisation of less complex flow characteristics such as figure 3, produce very understandable results from the different strategies employed.

When visualising more complex flow data such as the Bernard flow simulation figure 4, the issue of occlusion can significantly increase. The ability of the user to be able to filter specified surfaces from the rendering can reduce much of the clutter improving the overall visualisation.

The seeding strategy employed removes the need for the user to conduct lengthy examinations of the flow fields using manual seed placement techniques. The technique shows adequate domain coverage, and captures the features within the flow field for all the datasets we experimented with.

We have tested our algorithm on a variety of simulations, ranging from simple to complex including the simulation of a tornado, flow past a cuboid, Bernard flow as well as others. We found our visualisations were consistent with previous work and captured the same features.

5. Limitations

This is a work in progress paper. As such, the presented work has the following limitations:

1. Algorithm parameters require further analysis e.g. the choice of isolines and initial seeding curves need more exploration.
2. We would like to conduct more in-depth experiments on rendering surfaces with transparency.
3. We would like to analyse surface filtering in more depth.

The presented work provides inspiration for substantial future work.

6. Conclusions and Summary

We introduce a novel automatic method for the seeding of integral surfaces, and investigate a range of methods for the reduction of occlusion.

Despite the great amount of progress that has been made in the field of flow visualisation over the last two decades, a number of challenges remain. Challenges such as automatic path surface placement and perception remain key topics for further research.

Extending this algorithm to time-dependant flow fields is the main topic for further research.

References

[CCK07] CHEN Y., COHEN J. D., KROLIK J.: Similarity-Guided Streamline Placement with Error Evaluation. *IEEE Trans. Vis. Comput. Graph 13*, 6 (2007), 1448–1455. 2

[CSBI05] CAMARRI S., SALVETTI M. V., BUFFONI M., IOLLO A.: Simulation of the Three-Dimensional Flow Around a Square Cylinder Between Parallel Walls at Moderate Reynolds Numbers. In *Congresso di Meccanica ed Applicata* (sep 2005), pp. 11–15. 2

[GKT*08] GARTH C., KRISHNAN H., TRICOCHE X., TRICOCHE T., JOY K. I.: Generation of Accurate Integral Surfaces in Time-Dependent Vector Fields. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1404–1411. 3

[JL97] JOBARD B., LEFER W.: Creating Evenly–Spaced Streamlines of Arbitrary Density. In *Proceedings of the Eurographics Workshop on Visualization in Scientific Computing '97* (1997), vol. 7, pp. 45–55. 2

[Lar02] LARAMEE R.: Interactive 3D Flow Visualization Using a Streamrunner. In *CHI 2002, Conference on Human Factors in Computing Systems, Extended Abstracts* (April 20–25 2002), ACM SIGCHI, ACM Press, pp. 804–805. 2

[LGD*05] LARAMEE R. S., GARTH C., DOLEISCH H., SCHNEIDER J., HAUSER H., HAGEN H.: Visual Analysis and Exploration of Fluid Flow in a Cooling Jacket. In *Proceedings IEEE Visualization 2005* (2005), pp. 623–630. 2

[LGSH06] LARAMEE R. S., GARTH C., SCHNEIDER J., HAUSER H.: Texture-Advection on Stream Surfaces: A Novel Hybrid Visualization Applied to CFD Results. In *Data Visualization, The Joint Eurographics-IEEE VGTC Symposium on Visualization (EuroVis 2006)* (2006), Eurographics Association, pp. 155–162,368. 1

[LMGP97] LÖFFELMANN H., MROZ L., GRÖLLER E., PURGATHOFER W.: Stream Arrows: Enhancing the Use of Streamsurfaces for the Visualization of Dynamical Systems. *The Visual Computer 13* (1997), 359–369. 1

[LS07] LI L., SHEN H.-W.: Image-Based Streamline Generation and Rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (2007), 630–640. 2

[LWSH04] LARAMEE R., WEISKOPF D., SCHNEIDER J., HAUSER H.: Investigating Swirl and Tumble Flow with a Comparison of Visualization Techniques. In *Proceedings IEEE Visualization 2004* (2004), pp. 51–58. 2

[MCHM10] MARCHESIN S., CHEN C.-K., HO C., MA K.-L.: View-Dependent Streamlines for 3D Vector Fields. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (October 2010). (also Visualization 2010 Conference Proceedings). 2

[MLP*10] MCLOUGHLIN T., LARAMEE R. S., PEIKERT R., POST F. H., CHEN M.: Over Two Decades of Integration-Based, Geometric Flow Visualization. *Computer Graphics Forum (CGF) Vol. 29*, No. 6 (2010), pages 1807–1829. 2

[MT*03] MATTAUSCH O., THEUSSL T., , HAUSER H., GRÖLLER E.: Strategies for Interactive Exploration of 3D Flow Using Evenly-Spaced Illuminated Streamlines. In *Proceedings of the 19th Spring Conference on Computer Graphics* (2003), pp. 213–222. 2

[SLCZ09] SPENCER B., LARAMEE R. S., CHEN G., ZHANG E.: Evenly-Spaced Streamlines for Surfaces: An Image-Based Approach. *Computer Graphics Forum 28*, 6 (2009), 1618–1631. 2

[WHN*03] WEINKAUF T., HEGE H., NOACK B., SCHLEGEL M., DILLMANN A.: Coherent Structures in a Transitional Flow around a Backward-Facing Step. *Physics of Fluids 15*, 9 (September 2003), S3. Winning Entry from the Gallery of Fluid Motion 2003. 2

[WT02] WEINKAUF T., THEISEL H.: Curvature Measures of 3D Vector Fields and their Application. In *Journal of WSCG* (2002), V.Skala, (Ed.), vol. 10. 2

[ZSH96] ZÖCKLER M., STALLING D., HEGE H.: Interactive Visualization of 3D-Vector Fields Using Illuminated Streamlines. In *Proceedings IEEE Visualization '96* (Oct. 1996), pp. 107–113. 2