EG UK Theory and Practice of Computer Graphics (2012)
Hamish Carr and Silvester Czanner (Editors)

Advanced, Automatic Stream Surface Seeding and Filtering

M. Edmunds[1], R. S. Laramee[1], Guoning Chen[2], Eugene Zhang[3], and Nelson Max[4]

[1]Swansea University, UK. E-mail: {csmatti, r.s.laramee}@swansea.ac.uk.
[2]University of Utah, US. E-mail: chengu@sci.utah.edu
[3]Oregon State University, US. E-mail: zhange@eecs.oregonstate.edu
[4]University of California, Davis, US. E-mail: max@cs.ucdavis.edu

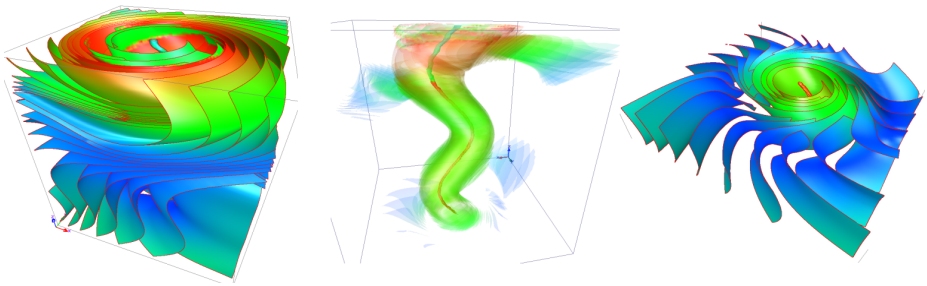


Figure 1: *Automatic Stream Surface Seeding: A set of stream surfaces seeded automatically using our technique on the* 128^3 *tornado simulation at time step zero. The left image shows surfaces visualized using silhouette edges. The center image shows surfaces with opacity mapped to the magnitude of local curl. Pixels are filtered according to opacity. This reduces occlusion and allows insight into the inner flow structures. The right image demonstrates the use of clipping planes.*

Abstract

The placement or seeding of stream surfaces in 3D flow fields faces a number of challenges. These challenges include perception, occlusion, and the appropriate representation of flow characteristics. A variety of streamline seeding approaches exist, little corresponding work is presented for stream surfaces. We present a novel automatic stream surface seeding and filtering algorithm. Our approach is designed to capture the characteristics of the flow utilizing illustrative techniques to alleviate occlusion and provide options for filtering. We define and prioritize a set of seeding curves at the domain boundaries from isolines computed from a derived scalar field. We detail the generation of an initial set of surfaces from the set of seeding curves, and discuss a technique for effective surface termination. We then present an algorithm that automatically seeds new interior surfaces, to represent locations not captured by the boundary seeding, at a user specified separation from the initial surface set. The results demonstrate satisfactory domain coverage and effective visualizations on a variety of simulations.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Flow Visualization, Stream Surface, Seeding Algorithm, Surface Filtering.

1. Introduction

Effective exploration and communication of fluid flow is a challenge. The breadth and depth of existing techniques differ in their complexity, quality and style [PVH*03] [LHZP07] [MLP*10]. Stream surfaces are an effective medium to convey not only the characteristics of the flow structures, but can also communicate additional information to the user. Stream surfaces for visualization face challenges such as occlusion and perception. A general solution to the problem of occlusion is to use transparency while illustrative techniques can be used to improve perception. Also filtering techniques may be utilized to reduce visual clutter. While surfaces offer many advantages in terms of information content, a basic visualization of the surface alone may not sufficiently provide information about the underlying data. For example a stream surface alone does not show the behaviour of any inner flow contained within the surface. Manual stream surface placement is based on trial and error, and important information can easily be missed. To maximize the information content for the user, stream surfaces must be carefully seeded and illustrated and thus providing strong motivation for studying stream surfaces and their seeding. We summarize the benefits and contributions of this paper here:

- A novel approach to seeding stream surfaces in 3D flow fields automatically seed surfaces throughout the domain based on user-specified separation.

- Effective boundary computed seeding curve prioritization to provide consistent and objective visualizations.
- Surface proximity termination based on distance field techniques.
- Techniques for surface filtering and illustration enhancing the information content.

A review of related literature is conducted in Section 2. A detailed presentation of the algorithm is given in Section 3. The results are discussed in Section 4. Conclusions and proposed future work are highlighted in Section 5.

2. Related Work

Starting with a definition of a streamline and a stream surface, this section discusses the related streamline and stream surface placement work. The section finishes with a discussion of related illustrative techniques.

A *streamline* is the trace of a massless particle from an initial location (seed point) and is tangent to the velocity field at every point along its length. If $\mathbf{v}(p)$ is a 3D vector field, the streamline through a point x_0 is the solution $I(x_0,t)$ to the differential equation:

$$\frac{d}{dt}I(x_0,t) = \mathbf{v}(I(x_0,t))$$

with the initial condition $I(x_0,0) = x_0$. A *stream surface* is the trace of a 1D seeding curve C through the flow. The resulting surface is everywhere tangent to the local flow. A stream surface S is defined by:

$$S(s,t) := I(C(s),t)$$

S is the union or continuum of integral curves passing through the seeding curve C. $S(s,t)$ is the surface parameterization where s is a streamline and t is a timeline.

Streamline Seeding Zöckler et al. introduce a method of illuminating streamlines [ZSH96]. For the placement of streamlines a stochastic seeding algorithm is applied. The degree of interest in each cell is based on a scalar value (i.e., velocity magnitude). See Weinkauf et al. [WHN*03] [WT02] 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.

The approach by Vilanova et al. [VBvP04] concentrates on seeding hyper-streamlines and the use of surfaces to visualize Diffusion Tensor Imaging data. 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 visualization. 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.

Methods described for seeding streamlines would not necessarily translate directly to seeding stream surfaces. Constructing a surface from discontinuous seed locations will produce inconsistent, twisted, folded, self occluded surfaces conveying little meaningful information. To produce well formed surface representations the seeding structure should be smooth; designed to represent some underlaying flow characteristic.

Placement Van Wijk [vW93] provides an alternative stream surface construction technique based on the global representation of the stream surfaces as implicit surfaces $f(x) = C$. The implicit method constructs surfaces where they intersect with the boundary. Theisel et al's approach to constructing saddle connectors in place of separating stream surfaces is an effort to address the challenges of occlusion [TWHS03].

This endeavor was extended by Weinkauf et al. [WTHS04] using separating stream surfaces originating from boundary switch curves. A *boundary switch curve* represents the location where outflow at the boundary plane switches to inflow. 2D separatrices originating from boundary switch points are studied by de Leeuw and van Liere [dLvL99].

Boundary switch curves prove to be a useful topological feature for generating separating stream surfaces partitioning the vector field into distinct regions of flow behavior. The limitation of boundary switch curves is that they may not exist at a boundary of the flow field. Our work extends this idea by generating seeding curves which uniformly partition the area between boundary switch curves (inclusive), and areas where they don't explicitly exist.

Peikert et al. [PS09] present topology relevant methods for constructing stream surfaces which visualize critical points, periodic orbits, unbounded curvature, and tightly winding spirals. The authors discuss error bounds and give application examples for the range of topological features under consideration. Edmunds et al. [EML*11] present a work-in-progress concept for generating seeding curves at the domain boundary. We present a complete study of the Edmunds et al. concept in this paper.

The approach reviewed in this subsection is placement of stream surfaces derived from some characteristic of the flow. With the exception of Peikert et al. the work focuses on capturing flow structures at the domain boundary. Peikert et al. discuss the construction of surfaces bounding topological structures.

Surface Rendering and Visualization Löffelmann et al. introduce methods for placing arrows on stream surfaces which improve perception of flow [LMGP97]. Loffelmann et al. [LMG97] then improve on this using hierarchical techniques to optimize the surface tiling. Laramee et al. [LGSH06] draw on previous image based texture advection research in order to improve the information content and perception of flow on stream surfaces.

Progress for the illustration of stream surfaces has been made by Born et al. [BWF*10] and Hummel et al. [HGH*10].

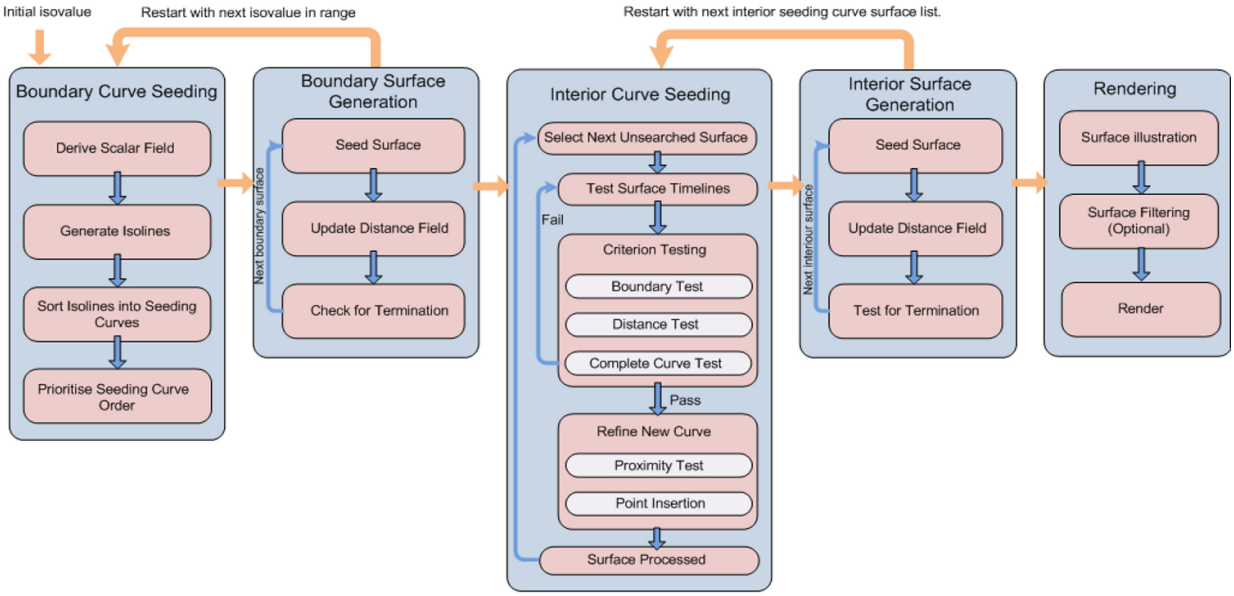


Figure 2: *Algorithm pipeline: The first stage creates a scalar field at the domain boundary Ω' based on the angle of incidence of out flow. A set of isolines is generated, then prioritized to create seeding curves for the stream surfaces. The stream surfaces are integrated through the velocity field. In parallel with surface advancement, a distance field, Ω_d, is updated describing the proximity to existing surfaces. This process is repeated for a range of isovalues. After generating an initial set of surfaces, new seeding curves are computed and refined in the domain interior Ω. Each surface, in a list of unprocessed surfaces, is searched along its length to locate timelines for the potential production of seeding curves. Timelines are offset at distance d_{sep} in the direction of the surface normal. If they pass a set of refinement criterion, they are used for the construction of smooth seeding curves. These seeding curves initialize new stream surfaces. This process is repeated until Ω is covered. The final stage filters and renders the surfaces using semi-opacity based on curvature, clipping, and color mapping.*

Born et al. describe techniques such as contour lines and halftoning to show the overall surface shape. Flow direction as well as singularities on the stream surface are depicted by illustrative surface streamlines. Hummel et al. examine how transparency and texturing techniques can be used with integral surfaces to convey both shape and directional information.

Illustrative techniques are important for the effective visualization of surfaces. Utilizing depth cues for depicting surface shape and color mapping and texturing to convey broader/clearer information content to the user, are examples of the advantages surface visualizations can provide.

3. Automatic Surface Seeding

This section presents the automatic stream surface seeding algorithm. We start with an overview of the seeding pipeline. Refer to Figure 2. The achievement of complete domain coverage in areas such as recirculating flow is the focus of the seeding strategy. Generation of surfaces in areas of flow which cannot be traced to the domain boundary is important to provide a complete visualization. The algorithm input is a 3D steady vector field $\vec{v}(\mathbf{p}) \in \mathbb{R}^3$ where $\vec{v}(\mathbf{p}) = [v_x(x,y,z), v_y(x,y,z), v_z(x,y,z)]$ defined for $\mathbf{p} \in \Omega$ and $\Omega \subset \mathbb{R}^3$

1. The starting point is the derivation of boundary seeding curves. We generate the seeding curves from isolines derived at the boundary of Ω denoted Ω'. The 2D scalar field $f(\Omega')$ is a function of flow direction exiting Ω. The isolines are constructed using marching squares. Once the vertices are correctly ordered and stored, the seeding curves are then prioritized into the optimal seeding order. Refer to Section 3.1.
2. Once the boundary seeding curves are computed, we generate stream surfaces advancing through Ω until they meet terminating conditions. The termination parameters are maximum length l_{max} and distance d_{test} to neighboring surfaces. A distance field is used to evaluate d_{test}. The process is repeated from step one until all isovalues are seeded. Section 3.2 and 3.3 provide further details.
3. We search the initial surface list along their length to locate empty regions. Each timeline of a given surface is offset normal to the surface d_{sep} . The proposed timeline is then analyzed for suitability. If the resultant curve does not pass the suitability criterion, we recursively select the next downstream timeline. Refer to Section 3.4.
4. The next step produces a smooth seeding curve from a suitable offset timeline. This refinement is achieved by removing vertices which are in close proximity to each other. Interpolating new vertices maintains an evenly distributed smooth discretised curve. Refer to Section 3.5.
5. The originating surface is transferred to the processed list and searched no further once a suitable seeding location is found. The new interior stream surface is added to the list of unprocessed surfaces. The surfaces are searched both

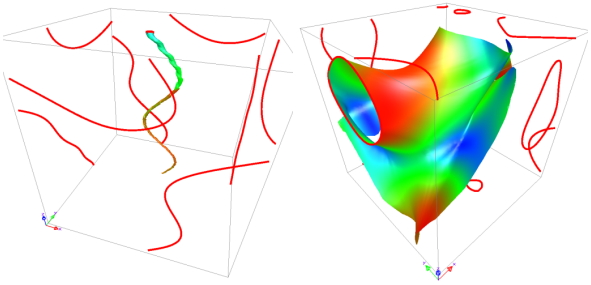


Figure 3: *Tornado and ABC simulations with boundary seeds generated using an isovalue of 0.9. In these figures a surface is generated from the first boundary seeding curve in the prioritized list of curves. The left image illustrates the prioritization capturing the vortex core of the Tornado. The right image shows the longest surface propagating from a closed loop boundary seeding curve.*

sides iteratively through the list of unprocessed surfaces until empty. Section 3.6 provides further details.

6. The final step is rendering the scene. A number of techniques are implemented to aid the viewer in perceiving the resulting visualization. This includes the use of transparency, color mapping, clipping planes, edge highlighting, and surface filtering. Refer to Section 3.7.

3.1. Boundary Seeding Curve Generation

For the problem of generating seeding curves between and inclusive of boundary switch curves, we compute a scalar field representing the flow exit trajectory from the domain boundary. When the flow is parallel to the domain boundary the scalar is one. When the flow exits orthogonal to the domain boundary the scalar is zero. This approach reduces the problem to a simple marching squares isoline extraction technique.

Generation of seeding curves from isolines derived from Ω' is performed in three steps. The first step is to define a scalar field. The derived field $f(\Omega')$ represents the scalar field which is a function of the direction of flow exiting Ω. The scalar represents the angle of incidence between the vector field defined at Ω' and the domain extent itself. The calculation is performed by projecting the unit vector onto a plane at Ω' (See Figure 4). The resultant magnitude $||\vec{v'}||$ 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 Ω' using a marching squares algorithm. The resulting vertices would normally be rendered as order independent line segments. However the vertices require sequential ordering for the seeding curve.

The range of isovalues used in the construction of the isolines, and resultant density of seeded surfaces, can be specified by the user. This effectively seeds surfaces between pairs of boundary switch curves, where they exist. The algorithm subdivides the range from 1 to 0 by a user defined division.

The seeding curves are prioritized in order of surface generation. The order of seeding new surfaces can influence the resulting visualization. A logical order is longest seeding curve first. This heuristic is based on the idea of prioritizing large domain filling surfaces. After some experimentation we find that seeding from closed loop curves is beneficial to visualizing vortex cores. The final heuristic is to seed closed loop curves first; longest to shortest, and then seed open ended curves; longest to shortest, starting with an isovalue nearest one; seeding each set of isolines in descending order of isovalue. Refer to Figure 3.

3.2. Boundary Surface Generation

Our work utilizes an out of the box solution to stream surface computation. Stream surfaces are propagated from each of the seeding curves defined in section 3.1. The surfaces are then terminated according to a set of conditions such as l_{max}, boundary proximity, and d_{test}.

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

3.3. Distance Field

The detection and computation of distance to the nearest object in a given domain from any given point is non trivial. A brute force approach can be implemented testing the distance to every vertex of every object in the domain with the current vertex and storing the shortest. This method is very expensive and thus distance field techniques to improve the speed are used.

In general, there are two groups of approaches [JBS06]: Distance computation for common surface representations, which discards most of the objects by exploitation of spacial coherency e.g. computing distances from objects in close proximity. Distance transforms are a second category which initially use a similar technique to evaluate distances to certain regions e.g. a thin layer around the surface, and then propagates them through Ω.

The most interesting method appropriate for use with discretized surfaces is the Vector City Vector Distance Transform

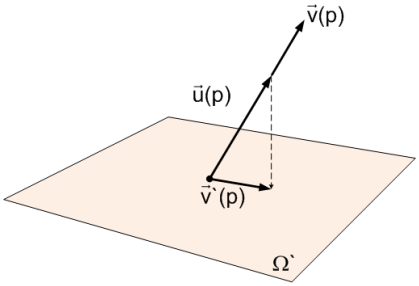
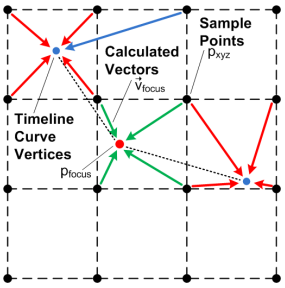
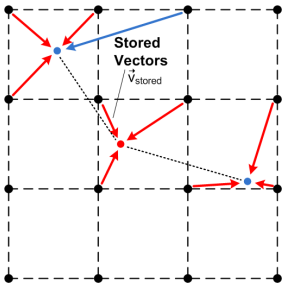


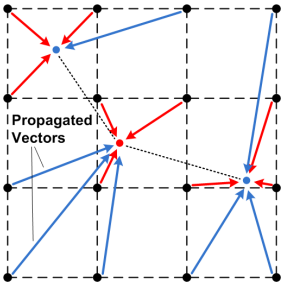
Figure 4: *Vector projection. $\vec{u}(p)$ is the unit vector of $\vec{v}(p)$, e.g. $\vec{u}(p) = \vec{v}(p)/||\vec{v}(p)||$. $\vec{v'}(p)$ is the vector projected onto Ω'.*



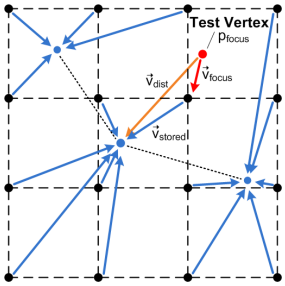
(a) For a given set of vertices where the focal vertex $\mathbf{p}_{focus}$ is highlighted in red, a set of vectors $\vec{v}_{focus}$ representing the distance from each of the vertices $\mathbf{p}$ of the bounding cell are calculated, i.e. $\vec{v}_{focus} = \mathbf{p}_{focus} - \mathbf{p}$.



(b) $||\vec{v}_{focus}||$ is compared to the currently stored vectors $\vec{v}_{stored}$, which are replaced if the magnitude is smaller, i.e $\vec{v}_{stored} = min(||\vec{v}_{stored}||, ||\vec{v}_{focus}||)$.



(c) The vector information is then propagated throughout the domain using the Vector City Vector Distance Transform (VCVDT) method which is detailed in [SJ01].



(d) Proximity is computed thus: $||\vec{v}_{dist}|| = ||\vec{v}_{focus} + \vec{v}_{stored}||$, where $\vec{v}_{focus}$ is the distance from the test vertex $\mathbf{p}_{focus}$ to the cell vertex $\mathbf{p}$.

Figure 5: *The distance field is a fast and versatile method used to locate the nearest object in the domain. The distance test compares the shortest magnitude to a predefined minimum allowable distance.*

or VCVDT [SJ01]. The first step is to calculate the vector $\vec{v}_{focus}$ from the vertex of interest $\mathbf{p}_{focus}$ (in our case surface vertices) to the bounding cell vertices $\mathbf{p}$. See Figure 5(a). The vector magnitude, $||\vec{v}_{focus}||$, is compared with the currently stored vector magnitude, $||\vec{v}_{stored}||$, at all bounding cell locations. The shortest in each location i.e. $||\vec{v}_{focus}|| < ||\vec{v}_{stored}||$ is stored replacing the original vector. See Figure 5(b). This process is repeated for every vertex representing the surface. Ω_d is then propagated throughout the domain. See Figure 5(c). While computing a new surface each new vertex is proximity tested against Ω_d. If the distance is too short $||\vec{v}_{dist}|| < d_{test}$ then the surface front propagation is terminated. See Figure 5(d). Once each surface is terminated it is then added to the distance field regardless of termination method.

3.4. Timeline-Based Seeding Curve Generation

This stage of the pipeline takes a list of stream surfaces seeded from Ω' from which additional surfaces can be seeded in order to gain complete domain coverage. This list is initially unprocessed. Each unprocessed surface is searched in turn to find a suitable location to generate a new seeding curve. This involves offsetting each timeline in turn along the length of the surface to find empty regions at which seeding of new interior stream surfaces can take place. Each timeline of a given stream surface is projected normal to the surface and analyzed for suitability against predefined criteria. The surfaces are searched both along their front and rear face.

The first criteria is the boundary test. Every new vertex of the candidate seeding curve must be inside Ω. Therefore any vertex $\mathbf{p}_i$ projected outside Ω is rejected (Figure 6(a)). The second criteria is the distance test. Each candidate curve vertex is tested against Ω_d. The vertex is marked reject if the location is within a user defined minimum distance parameter d_{test}, e.g. too close to a neighboring surface (Figure 6(b)).

The candidate curve is then tested for spacial continuity guided by user defined parameters of minimum contiguous length d_{length}, and maximum split distance d_{split}. If any resulting gaps in the offset timeline are too large, or any remaining section is too short, the candidate seeding curve is rejected. If the candidate curve does not pass the criterion then the next timeline along the length of the surface is selected, and the process repeats.

3.5. Interior Seeding Curve Refinement

If the candidate interior seeding curve passes the previously described criterion it is refined. The refinement process is intended to evenly distribute and smooth out any inconsistencies with the new curve. As a result of the projection in concave or convex areas of the surface the vertices may be too close to one another, too far apart.

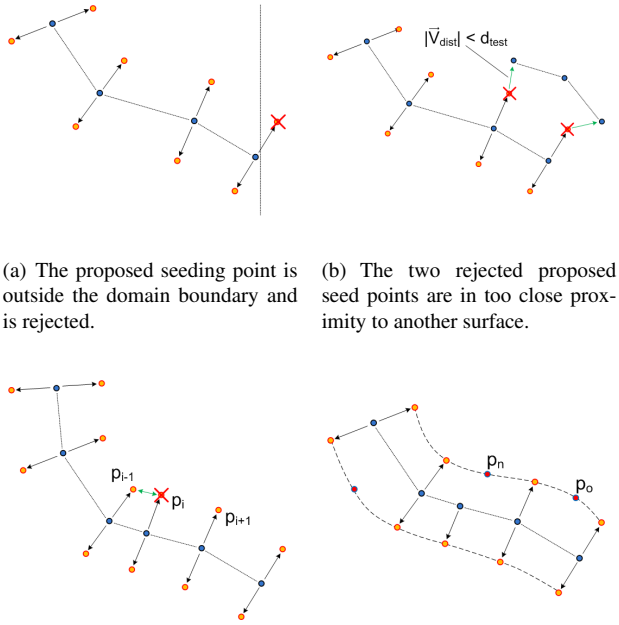
The refinement starts by removing vertices too close in proximity. The vertices are tested in groups of three, $\mathbf{p}_{i-1}$, $\mathbf{p}_i$, $\mathbf{p}_{i+1}$, starting from one end of the curve, incrementing one vertex at a time. The purpose of this approach is to test the central vertex against a user defined proximity d_{prox} to its neighbors. If a neighboring vertex is too close $||\mathbf{p}_{i-1} - \mathbf{p}_i|| < d_{prox}$ or $||\mathbf{p}_i - \mathbf{p}_{i+1}|| < d_{prox}$, vertex $\mathbf{p}_i$ is removed (Figure 6(c)).

The curve is further refined by inserting additional vertices at locations where proximity to the next vertex is too great. The insertion process uses cubic interpolation (Catmull-Rom spline). The process of insertion marks the position along the curve a new vertex $\mathbf{p}_n$ should be inserted. A new vertex is interpolated and held in a temporary list. This is repeated for every section along the curve (Figure 6(d)).

Once completed the temporary list of vertices, $\mathbf{p}_n$ and $\mathbf{p}_o$ in the example, are then inserted at the correct locations along the seeding curve. The approach of inserting the vertices into the array post interpolation prevents newly inserted vertices from violating the proximity test, interfering with the interpolation. This process is repeated for all candidate interior seeding curves.

3.6. Interior Stream Surface Generation

Once the seeding curve has been formed, a new surface is generated in both downstream and upstream directions. Distance



(a) The proposed seeding point is outside the domain boundary and is rejected.

(b) The two rejected proposed seed points are in too close proximity to another surface.

(c) The proposed seed points are too close and is rejected.

(d) New vertices are inserted into the seeding curves.

Figure 6: *The process of accepting or rejecting proposed new interior seeding points.*

field Ω_d is updated with the vertices representing the new surface. The given stream surface is added to the list of processed surfaces. The new surface is added to the list of unprocessed surfaces. This is repeated for every unprocessed surface until no further interior seeding curves, and therefore surfaces, can be generated. The surface termination criteria are the same as for boundary surfaces.

3.7. Stream Surface Filtering and Rendering

A number of techniques are implemented to aid the viewers perception of the visualization. The techniques used to render the results include the use of transparency, color, silhouette edge highlighting, lighting and shadow, and surface filtering.

Color is mapped to $||\vec{v}||$ and opacity is mapped to the magnitude of vector field curl $||\nabla \times F||$. The curl of vector field F is described as a vector having magnitude equal to the circulation at each point of F, and is perpendicular to the plane of circulation at each point. In Cartesian coordinates this is defined as:

$$\nabla \times F = \left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}\right)\hat{x} + \left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}\right)\hat{y} + \left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right)\hat{z}$$

Lighting and shading are standard tools for depth and shape perception. Silhouette edge highlighting is used to help the viewer understand where the surfaces curve away from any given viewpoint, and enhances the perception of surface edges.

Another technique involves filtering of the surfaces to aid in the reduction of visual clutter. As our visualizations are rendered with opacity mapped to $||\nabla \times F||$, the result is opaque surfaces in areas of high vector field curvature i.e., opaque vortex cores. This criterion is defined on a normalized scale which can interactively be adjusted by the user. An example of this can be seen in Figure 1 (middle).

4. Results

We achieve full domain coverage and capture the properties of the flow field by allowing the user to control the density of neighboring stream surfaces. Figure 1 shows results from automatic seeding in a tornado simulation. The surfaces are seeded using boundary seeding from isovalue 1.0 to 0.0 at 5 intervals. It can be seen that the domain is adequately covered to capture the structure of the tornado. On any planar domain boundary, the scalar field range may not extend to 1 (boundary switch curve). However, our method is able to generate seeding curves at isovalues less than 1 effectively extrapolating between boundary switch curves on an extended boundary plane. Using transparency, filtering and silhouette edges improve the users perception of the results. The center image highlights well the tornado core using curl based surface filtering.

Figure 7 demonstrates the interior seeding and resultant surfaces filling the domain from an initial set of surfaces generated at the boundary. The bottom image uses a clipping plane to section the flow field. It clearly shows the interior seeding following the structure of the tornado. Figure 8 visualizes the vortices generated behind the cuboid. A section through the visualization aids the users perception of these characteristics.

The Lorenz attractor can be seen in Figures 9 and 10. The images in Figure 9 show the vector field visualized with a sparse set of surfaces. The top left image is seeded only from the boundary. Using interior seeding the top right image shows complete domain coverage. The bottom image uses clipping

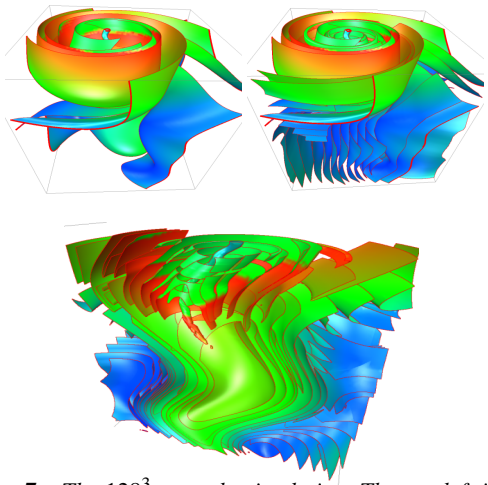


Figure 7: *The* 128^3 *tornado simulation. The top left image shows surfaces seeded with boundary seeding using an isovalue of 0.9. The boundary seeding curves are highlighted in thick red. The top right image shows seeding interior surfaces at* $d_{sep} = 5.0$*. The bottom image shows the final visualization with silhouette edges and clipping planes.*

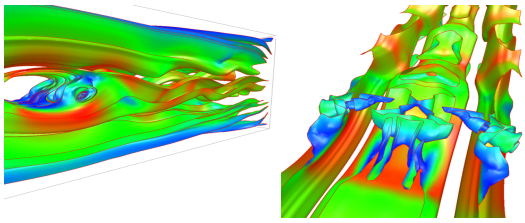


Figure 8: *This is a direct numerical Navier Stokes simulation by Simone Camarri and Maria-Vittoria Salvetti (University of Pisa), Marcelo Buffoni (Politecnico of Torino), and Angelo Iollo (University of Bordeaux I) [CSBI05] which is publicly available [Int]. We use a uniformly resampled version which has been provided by Tino Weinkauf and used in von Funck et al. for smoke visualizations [vFWTS08]. The images show a set of 5 isovalues used to fill the domain. Color is mapped to velocity. The left image visualizes the generated surfaces and demonstrates domain coverage. The right image uses edge highlighting and is clipped to revel the center of the visualization. Note: The cuboid is not shown for clarity.*

planes to understand the inner flow structure. The second set of images in Figure 10 demonstrates a denser visualization which better captures the underlying flow characteristics using the same techniques as Figure 9. The ABC visualization shown in Figure 11 demonstrates the capture of the inner flow structures. The use of clipping planes and edge highlighting significantly improve the perception of the flow field. The illustrative strategies implemented for resolving clutter, resulting from seeding multiple surfaces, improve perception and therefore aid understanding of the underlying flow structures. Visualization of less complex flow characteristics produce clear results from the different strategies employed. When visualizing more complex flow data enabling the user to filter surfaces (Figure 1 middle) or clip areas of the domain (Figure 11 right) can reduce the visual clutter improving the information content of the visualization. In practical applications indicating the flow direction would by required by the user. This can be implemented using textures applied to the surface mesh, or using techniques such as [CSFP12] by Carnecky et al. Also, stream surface generation can be influenced by topological structures such as invariance and separation features and is discussed in depth by Schneider et al. [SRWS10].

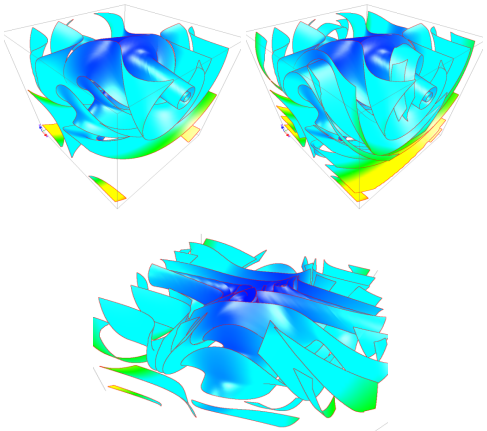


Figure 9: *The Lorenz attractor with a resolution of* 128^3. *The parameters for this simulation are;* $\sigma = 10$, $\rho = 28$, *and* $\beta = 8/3$. *The familiar circulation interaction is seen from the underside of the domain. This visualization shows a spars set of seeding parameters. The top left image shows boundary seeding at 3 intervals. The top right image shows the interior seeding filling the remaining unseeded areas of the domain using* $d_{sep} = 10.0$. *The bottom image is clipped revealing the inner flow characteristics.*

5. Conclusions and Summary

We introduce a novel automatic method for the seeding of stream surfaces. We investigate a range of methods for improved perception, information content and reduced clutter. The seeding strategy employed removes the need for the user to conduct lengthy examinations of the flow fields using manual seed placement techniques. We tested our algorithm on a variety of simulations, ranging from simple to complex. The techniques show adequate domain coverage capturing the features within the flow field.

The limitation of this technique is it cannot be applied to a domain which only parallel flow exists entering or exiting the domain. This is also true if there are no inflow/outflow regions at the domain boundary.

However, this method is easily extendable to a cube shaped seeding structure residing within the domain boundary. There is also potential to extend this algorithm to path and streak surface seeding. Perception and information content remain key topics for further research.

6. Acknowledgments

The Authors would like to thank the Department of Computer Science at Swansea University, UK, the Department of Computer Science at the University of Utah, US, and the Department of Computer Science at Oregon State University, US. Guoning Chen was supported by DOE SciDAC VACET.

References

[BWF*10] BORN S., WIEBEL A., FRIEDRICH J., SCHEUERMANN G., BARTZ D.: Illustrative Stream Surfaces. *IEEE Transactions*

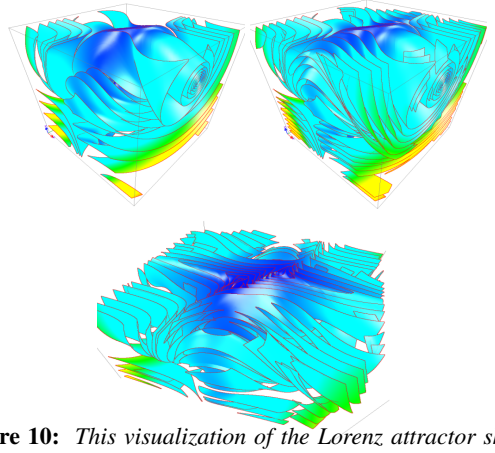


Figure 10: *This visualization of the Lorenz attractor shows the effect of denser seeding. The top left image shows boundary seeding at 5 intervals. The top right image shows interior seeding using* $d_{sep} = 5.0$. *The bottom image is clipped revealing the inner flow characteristics.*

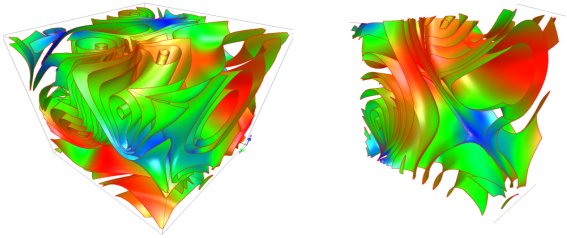


Figure 11: *A 128^3 Arnold-Beltrami-Childress (ABC) simulation. The left image shows boundary seeding at 5 intervals. The images show surfaces with color mapped to velocity. The right visualization is sliced using clipping planes.*

on Visualization and Computer Graphics 16, 6 (2010), 1329–1338. 2

[CCK07] CHEN Y., COHEN J. D., KROLIK J.: Similarity-Guided Streamline Placement with Error Evaluation. *IEEE Transactions on Visualization and Computer Graphics 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 *XVII Congresso di Meccanica Teorica ed Applicata* (2005). 7

[CSFP12] CARNECKY R., SCHINDLER B., FUCHS R., PEIKERT R.: Multi-layer Illustrative Dense Flow Visualization. *Computer Graphics Forum 31*, 3 (2012), 895–904. 7

[dLvL99] DE LEEUW W. C., VAN LIERE R.: Collapsing Flow Topology Using Area Metrics. In *Proceedings IEEE Visualization '99* (1999), pp. 349–354. 2

[EML*11] EDMUNDS M., MCLOUGHLIN T., LARAMEE R. S., CHEN G., ZHANG E., MAX N.: Automatic Stream Surfaces Seeding. In *EUROGRAPHICS 2011 Short Papers* (Llandudno, Wales, UK, April 11–15 2011), pp. 53–56. 2

[HGH*10] HUMMEL M., GARTH C., HAMANN B., HAGEN H., JOY K.: IRIS: Illustrative Rendering for Integral Surfaces. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1319–1328. 2

[Int] International CFD Database, http://cfd.cineca.it/. 7

[JBS06] JONES M. W., BAERENTZEN J. A., SRAMEK M.: 3d distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics 12* (2006), 581–599. 4

[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

[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. 2

[LHZP07] LARAMEE R., HAUSER H., ZHAO L., POST F. H.: Topology-Based Flow Visualization: The State of the Art. In *Topology-Based Methods in Visualization (Proceedings of Topo-in-Vis 2005)* (2007), Mathematics and Visualization, Springer, pp. 1–19. 1

[LMG97] LOFFELMANN H., MROZ L., GROLLER E.: *Hierarchical Streamarrows for the Visualization of Dynamical Systems*. Technical report, Institute of Computer Graphics, Vienna University of Technology, 1997. 2

[LMGP97] LOFFELMANN H., MROZ L., GROLLER E., PURGATHOFER W.: Stream Arrows: Enhancing the Use of Stream-surfaces for the Visualization of Dynamical Systems. *The Visual Computer 13* (1997), 359–369. 2

[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

[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 (2010). 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 29*, 6 (2010), 1807–1829. 1

[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

[PS09] PEIKERT R., SADLO F.: Topologically Relevant Stream Surfaces for Flow Visualization. In *Proc. Spring Conference on Computer Graphics* (April 2009), Hauser H., (Ed.), pp. 43–50. 2

[PVH*03] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: The State of the Art in Flow Visualization: Feature Extraction and Tracking. *Computer Graphics Forum 22*, 4 (Dec. 2003), 775–792. 1

[SJ01] SATHERLEY R., JONES M. W.: Vector-city vector distance transform. *Computer Vision and Image Understanding 82*, 3 (2001), 238–254. 5

[SRWS10] SCHNEIDER D., REICH W., WIEBEL A., SCHEUERMANN G.: Topology Aware Stream Surfaces. In *Eurographics/IEEE Symposium on Visualization* (Bordeaux, France, June 9–11 2010), vol. 29, pp. 1153–1161. 7

[TWHS03] THEISEL H., WEINKAUF T., HEGE H.-C., SEIDEL H.-P.: Saddle Connectors–An Approach to Visualizing the Topological Skeleton of Complex 3D Vector Fields. In *Proceedings IEEE Visualization '03* (2003), pp. 225–232. 2

[VBvP04] VILANOVA A., BERENSCHOT G., VAN PUL C.: DTI visualization with streamsurfaces and evenly-spaced volume seeding. In *Joint Eurographics - IEEE TCVG Symposium on Visualization* (2004), Deussen O., Hansen C., Keim D., Saupe D., (Eds.), Eurographics Association, pp. 173–182. 2

[vFWTS08] VON FUNCK W., WEINKAUF T., THEISEL H., SEIDEL H.-P.: Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2008) 14*, 6 (November - December 2008), 1396–1403. 7

[vW93] VAN WIJK J. J.: Implicit Stream Surfaces. In *Proceedings of the Visualization '93 Conference* (Oct. 1993), IEEE Computer Society, pp. 245–252. 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, pp. 507–514. 2

[WTHS04] WEINKAUF T., THEISEL H., HEGE H. C., SEIDEL H.-P.: Boundary Switch Connectors for Topological Visualization of Complex 3D Vector Fields. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym 04)* (2004), pp. 183–192. 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