A Flow-guided Streamline Seeding Strategy

Vivek Verma[1], David Kao[2], and Alex Pang[1]
[1] Computer Science Department, UCSC
[2] NASA Ames Research Center
vivek@cse.ucsc.edu, davidkao@nas.nasa.gov, pang@cse.ucsc.edu
www.cse.ucsc.edu/research/avis/seed.html

Abstract

This paper presents a seed placement strategy for streamlines based on flow features in the data set. The primary goal of our seeding strategy is to capture flow patterns in the vicinity of critical points in the flow field, even as the density of streamlines is reduced. Secondary goals are to place streamlines such that there is sufficient coverage in non-critical regions, and to vary the streamline placements and lengths so that the overall presentation is aesthetically pleasing (avoid clustering of streamlines, avoid sharp discontinuities across several streamlines, etc.). The procedure is straight forward and non-iterative. First, critical points are identified. Next, the flow field is segmented into regions, each containing a single critical point. The critical point in each region is then seeded with a template depending on the type of critical point. Finally, additional seed points are randomly distributed around the field using a Poisson disk distribution to minimize closely spaced seed points. The main advantage of this approach is that it does not miss the features around critical points. Since the strategy is not image-guided, and hence not view dependent, significant savings are possible when examining flow fields from different viewpoints, especially for 3D flow fields.

Key Words and Phrases: seed placement, streamline, critical point, Voronoi diagram, Poisson disk distribution.

1 INTRODUCTION

There are a number of methods for streamline placement that mostly address the aesthetic aspects of a flow visualization using streamlines. These methods [11, 16] describe how the streamlines should be placed in a flow field so that the visualization does not appear to be cluttered and there are no artifacts introduced in the visualization process that might lead to a misinterpretation of the flow field. In our work we address an important issue that has been largely neglected by these methods. Namely, whether the streamlines placed by these methods result in a visualization that captures all the important features (e.g. critical points) of the flow field. Our streamline seeding strategy guarantees that important features like critical points are not missed. If the streamlines are not seeded appropriately (e.g. using regular or random seeding), or using image-guided streamline placement alone, important details of the flow can be missed. This problem is illustrated in Figure 1. Figure 1a shows streamlines generated using a regular seeding strategy, Figure 1b shows the result of using image-guided streamline placement, while Figure 1c shows the result of using our seeding strategy. We can see that without proper seed placement, some details of the flow can be missed by the streamline visualization. The saddle critical point is not sufficiently captured by the streamlines in Figure 1a and Figure 1b. Streamlines generated using our method adequately highlights the critical points as shown in Figure 1c.

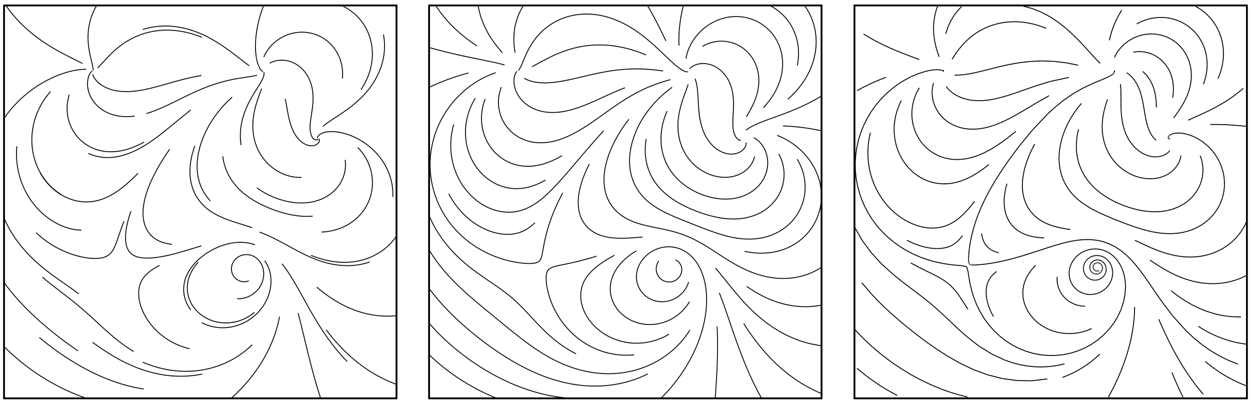

Figure 1: (a) Effects of regular seeding (49 streamlines), (b) effects of image-guided seeding (47 streamlines), (c) effects of flow-guided seeding (47 streamlines). Regular and image-guided seeding strategies may miss important flow features, specially when seeding is sparse. Both image-guided and flow-guided streamlines were generated such that the minimum separating distance of streamlines is 3% of the image width.

There are some important goals to consider in order to generate an effective streamline visualization. In particular, a good seeding strategy should have the following characteristics:

- *Coverage*: The streamlines should not miss any interesting regions in the vector field. The interesting regions are those that we would like to study in the vector field, e.g. critical points, separation, and re-attachment lines. In addition, streamlines should cover the entire region of the field. Hence, even if the field is more or less uniform in a region, some streamlines should indicate the uniform nature of the flow in these regions. This goal is easier to achieve than other goals because one can always generate a lot of streamlines such that nothing important is missed. However, simply populating the field with more streamlines is not acceptable because some areas in the flow field, such as convergent regions, will force streamlines to cluster together, making it difficult to distinguish among individual streamlines. More importantly, it defeats the characteristic of uniformity as described next.

- *Uniformity*: The streamlines should be more or less uniformly distributed over the field. This is a more challenging goal to achieve because while we can control where to place the seeds, we do not know how the resulting streamlines will behave. Uniformity is directly related to the density of streamlines crossing a unit area of the flow field. Hence, density of streamlines is an important parameter.

- *Continuity*: It is desirable from the point of view of aesthetics that the streamlines show continuity in the flow. Hence, one would prefer fewer long streamlines over many short streamlines. The latter tend to give the impression of "choppiness" while the former tend to give an impression of smooth continuous flow. In general, the longer the streamlines, the higher the likelihood that they will tend to crowd together in some arbitrary flow field. Therefore, this parameter needs to be balanced against the uniformity criterion.

Since most flow fields are defined over a grid, a popular seeding strategy is to seed at the grid points so that no important features are missed. This is usually an overkill and requires that more streamlines be traced than is necessary to capture all the desired details of the flow. Furthermore, the streamlines tend to clutter in ways that is difficult to predict. Even if the grid is sub-sampled to reduce the density of streamlines, cluttering is still difficult to avoid. Finally, regular seeding may also produce visualization artifacts that are not present in the flow field. Figure 2 shows streamlines with regular seed placement for two datasets. These images show that the streamlines placed on a regular grid can generate artifacts because the underlying regular grid can be perceived in the visualization (left image in Figure 2) and also create clutter if the streamlines are too long (right image in Figure 2).

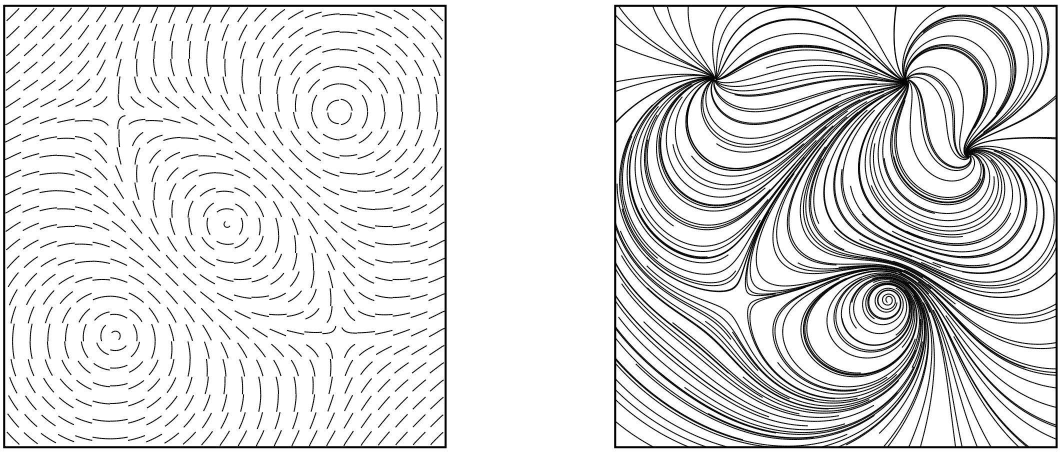

Figure 2: Streamlines are seeded on a regular grid for these two datasets. Left: the underlying regular grid can be perceived in the visualization. Right: streamlines can create clutter if their lengths are long.

Cluttering is of course dependent on the flow field. Blindly seeding on a regular grid results in a streamline visualization where the individual streamlines can be difficult to distinguish in important regions (e.g. regions where a critical point is present). If one does not seed all the grid locations and seed only every other grid point (for instance) then the streamlines might miss some interesting features. Regular seeding on a grid does not satisfy the requirements of *coverage* and *uniformity* for a good streamline visualization.

Image-guided techniques focus on the problem of cluttering. These methods also enforce a uniform spatial distribution of streamlines. However, they ignore coverage criterion, a scientifically important aspect of the flow visualization. The work presented in this paper attempts to alleviate this problem. We consider the coverage goal to be of greatest importance because from a scientific point of view the information content of any visualization is the most important. The goal of a uniform spatial distribution of streamlines is important only to the extent that it does not interfere with the most important goal of achieving a good coverage. The goal to achieve an esthetically pleasing visualization has its merits but it should not compromise the other two goals (coverage and uniformity), hence it is low on our priority list. Another problem with image-guided techniques is that they are view dependent and do not permit interactive manipulation of viewpoint.

Our approach starts with the assumption that if we know how the flow behaves and the location of the important features in the flow are known, then we can place seeds to trace streamlines more cleverly than the naive approach to place seeds at the grid locations, or to use an optimization strategy to reduce clutter. In fact, there has been significant research done to extract the important features we are interested in looking at. See [3, 7] for a definition and classification of critical points. These points are also used extensively in topological presentation of flow fields [7, 8, 9]. In this paper, we take advantage of the knowledge about the flow features in deciding how to place streamlines more intelligently. In particular, the type and location of the critical points are used to design and orient seeding templates that capture the flow patterns of these critical points. We are confident that our strategy can be easily extended to use information about other features like flow separation and re-attachment lines, flow topology lines, etc. to further improve the flow-guided seed placement for streamlines. For example, information about automatically extracted separation and attachment lines [12] can be used to improve seeding to highlight those regions.

2 RELATED WORK

There are very few methods that address the problem of generating visualizations with good streamline placement strategies. It should be noted that placing seeds uniformly does not result in uniformly spaced streamlines. Hence methods like Dovey's [5] to generate a uniform density of glyphs are not very useful for streamline placement. Max *et al.* [14] use particle traces on a 3D surface that are terminated when they come too close to the path of other particles. Turk and Banks [16] use the minimization of an energy function to guide the placement of streamlines at a specified density. Their method uses a low-pass filtered version of the current image to measure the difference between the current image and the desired density value. The energy is reduced iteratively by changing the positions and lengths of streamlines, merging streamlines, and creating new streamlines. The resulting placement has a hand-placed appearance and the streamlines appear to be neither too sparse nor too crowded. Computation time for their method is significant. Jobard and Lefer's method [11] creates evenly spaced streamlines that match the quality of streamlines generated using the image guided approach of Turk and Banks [16]. Recently, Mao *et al.* [13] have extended the image-based method of Turk and Banks to place streamlines on curvilinear grid surfaces. A major drawback of all these methods is that they do not take guidance from the important features of the flow. None of the existing methods for streamline placement guarantee that the resulting streamlines would capture all the essential features of the flow field. If the streamline separation distance chosen in Jobard and Lefer's method is not small enough, the streamlines could miss a critical point or the critical point might not be sufficiently captured by the streamlines. Our strategy guarantees that all the essential features will be captured by the resulting streamlines because we use the information about the location and type of critical points to seed the streamlines.

3 FLOW-GUIDED STREAMLINE PLACEMENT

Before we begin discussing our strategy to place seeds in a flow field, let us look at the different types of critical points in 2D flows. Figure 3 shows the different types of critical points that can be found in 2D flows.

If the flow field contains only one critical point (of any type) and if we know the location and nature of the critical point, then we can easily decide where to place the seeds to trace streamlines. Figure 4 illustrates the seeding pattern for different types of critical points. We have studied the different critical point types and have come up with the following strategy to place seeds in the vicinity of the critical points so that the streamlines traced from these seeds bring out the nature of the flow around the critical points. We call these seeding patterns to be *seed templates* for critical

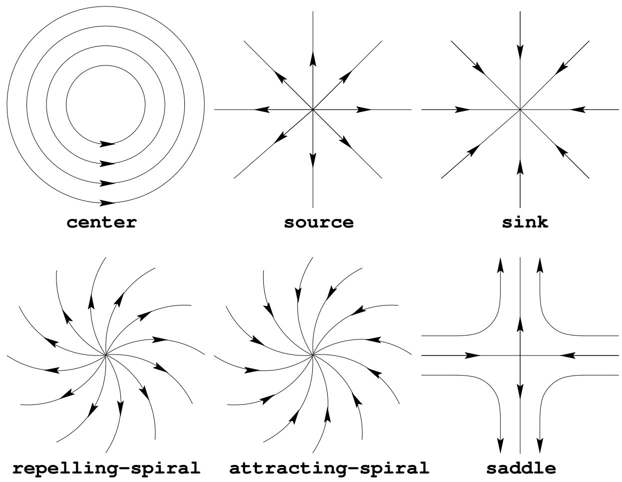

Figure 3: Different type of critical points possible in 2D flows.

points.

1. *center, spiral*: place seeds along a straight line emanating from the critical point location. Figure 4a shows the seed template for center and spiral type of critical points.

2. *source, sink*: place seeds along the perimeter of a circle around the critical point. Figure 4b shows the seed template for this type of critical point.

3. *saddle*: place seeds along the lines that bisect the principal eigen directions. Figure 4c shows the seed template for saddles.

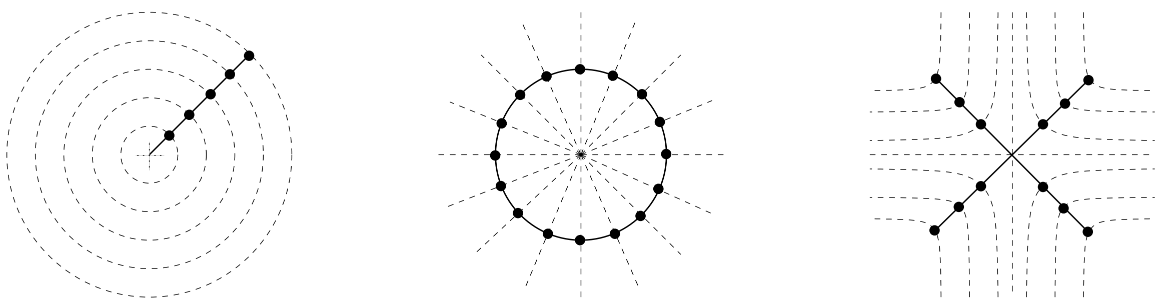

Figure 4: Seed templates for various critical points. The bold dots represent the seed template and the dashed lines are the streamlines traced using the seeds from the template. (a) center, spiral; (b) source, sink; (c) saddle.

In the following discussion, we will refer to the flow patterns around critical points, as depicted in Figure 3, as being ideal. That is, the flow pattern is representative of the flow that one might observe in the vicinity of these types of critical points. The flow pattern is the identifying characteristic that distinguishes one type of critical point from another. For example, we expect to see rotating flow around a center type of critical point.

An important parameter to decide for placing seeds along the template is the proximity of the seeds to the critical point. We have used some heuristics to decide how to choose the size of the template for a given critical point and other parameters like the distance between adjacent seeds on a template. The details of parameter selection are discussed in sections 3.1 and 3.2.

The view of flow fields presented above is however very simplistic. In general, multiple critical points may be present in the flow and they interact with each other to give rise to patterns that deviate from the above presented simplistic notion of a flow field. Looking at Figure 5, we note that the flow near critical points is very similar to the ideal flow pattern for that type of critical point. As we go further and further from a critical point location, the flow pattern is influenced by other critical points. That is, the ideal flow pattern is most prominent in the immediate vicinity of the critical point. Consequently, for any general flow, we can always find a neighborhood around each critical point where the flow behaves as if other critical points were not present in the flow. If we can find this neighborhood then it will be easier to place seeds close to the critical points for a streamline visualization that highlights the critical points "nicely". Hence, we proceed to determine a suitable partition of the flow such that only one critical point lies inside each partition. We have found that such a partition can be approximated by the Voronoi diagram constructed using all the critical point locations of the flow field.

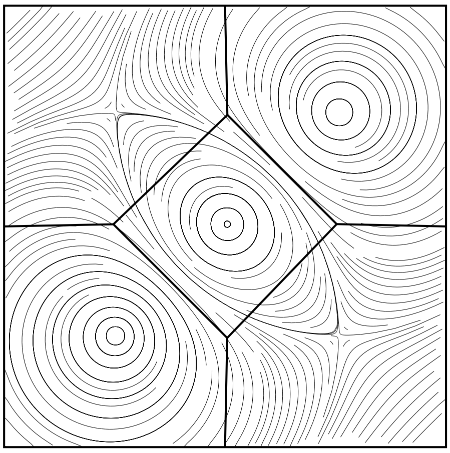

Figure 5: A flow can be partitioned using Voronoi diagram of the critical point locations. Each Voronoi region contains only one critical point that represents the flow in that region.

Given a set S of n distinct points in R^d, the Voronoi diagram is the partition of R^d into n polyhedral regions $vo(p)$, $(p \in S)$. Each region $vo(p)$, called the Voronoi region of p, is defined as the set of points in R^d which are closer to p than to any other point in S [6]. For our purposes, the set S is the set of critical point locations in R^2. Since some of the Voronoi regions will be unbounded, we also compute the intersection of the Voronoi diagram with the flow data's grid boundary. We chose to use Voronoi partitioning because the Voronoi regions are convex polygons with the property that every point in the Voronoi region around a critical point is closer to that critical point than to any of the other critical points. This means that the flow at the points in each Voronoi region is primarily influenced by the critical point it contains. This is an important heuristic because we have observed that the flow pattern at any point is influenced by its proximity to a critical point. The size of each Voronoi region is an approximation of the extent of the influence of the critical point it contains. In the following discussion, the boundary of a Voronoi region will be called its *Voronoi boundary.*

Figure 5 shows a Voronoi partition of the dynamic vortices dataset. Notice that each Voronoi region contains exactly one critical point and the flow within a region is characterized by the type of critical point it contains. The Voronoi regions are an approximation of the boundaries that partition the flow into different regions of flow types. Also important to notice is that around each critical point the flow would be close to ideal for that type of critical point. However, as you move away from the critical point, the flow will start to show influences from the flow around the neighboring critical points. It seems that the only critical points that might influence the flow near the boundary of a Voronoi region are those that are its neighbors because the flow patterns near the edges of the Voronoi regions depend on the critical points on both sides of each edge.

Given the above background, we can now outline a strategy to place seeds based on critical point location and types:

1. compute critical point locations and determine their types.
2. compute Voronoi partition of the set of critical points.
3. use template patterns around each critical point to place seeds and trace streamlines from these seeds.
4. place some "random" seeds to fill blanks.

We use FAST [1] to compute the critical point locations and to classify them. To compute the Voronoi diagram of the set of critical point locations, we use Jonathan Shewchuk's publicly available software called *triangle* [15]. For step 3 in the procedure described above, we need to make the following decisions:

- How big should the template pattern be?
- How far apart should the seeds be placed in the template?
- How long should the streamlines be?

We will discuss these issues in sections 3.1, 3.2, and 3.3, respectively. After seeding streamlines using templates for critical points, we add some random seeds that are distributed according to Poisson disk distribution. There are two important consequences of seeding with the seed template before the random seeds. By giving priority to seed templates, we ensure coverage of flow patterns near critical points. Furthermore, because we terminate a streamline when it comes close to an existing streamline (see Section 3.3), earlier streamlines will tend to be longer than later streamlines. Hence, streamlines traced from the seed templates are longer than those traced using seeds placed randomly to fill in blank spaces (see Section 3.4). Such a strategy ensures that the regions in the flow field close to critical points are given more importance than other regions.

3.1 Deciding size of seed templates

We are assuming that the flow inside each Voronoi region is characterized primarily by the type of critical point it contains. This means that each critical point should have its template constrained to lie completely within that critical point's Voronoi region.

We use the following strategy to decide the size of the seed templates for the various critical points.

- *center, spiral*: For center and spiral type of critical points, we find the line segment that joins the critical point to the closest point on the Voronoi boundary and seed along this line segment. One might ask what is so special about this particular line segment. We could have chosen many other line segments, for instance one that joins the critical point to the farthest point on the Voronoi boundary. We have experimented with many such possibilities and most of them resulted in too many streamlines and hence clutter. Basically, the ideal flow pattern of these critical points fade rather quickly.

- *source, sink*: For a source and sink type of critical points, we seed along a circle's perimeter. This circle has its center at the critical point and we chose it to be the largest circle that would fit completely inside the critical point's Voronoi region. Hence, the radius of this circle is equal to the distance between the critical point and the closest point on the Voronoi boundary. In contrast to centers and spirals, the ideal flow pattern of sources and sinks seem to extend further out.

- *saddle*: For a saddle we place seeds along two lines. These lines are the bisectors of the principal eigen vector directions. The extent of these lines is decided by their intersection with the Voronoi boundary. We have found that the saddles are the trickiest to seed because if the the seed closest to the saddle's location along the bisectors is not close enough then the saddles are not captured properly. For this reason, we decided to seed two special streamlines very close to the saddle. The seeds for these two streamlines are chosen to lie on the same bisector but on the opposite sides of the saddle's center. The distance of these special seeds from the center is chosen to be equal to one half the cell size of the grid. An additional note is that the seed templates for saddles presented in this paper assume index zero saddles which result in separation of flow into 4 regions around the critical point. Other types of saddles, e.g. index of -1, may result in more than 4 flow regions around the saddle point, and would require a different type of seed template. However, they would still be based on bisector lines.

The size (i.e. extent) of the seed template is adjustable. In practice, we find that half of full size of the seed template is sufficient because the observed flow patterns are ideal only in the vicinity of the critical points. This means that for source and sink critical points, we seed along the perimeter of a circle centered at the critical point. The radius of this circle is equal to half the distance of the critical point to the closest point on the Voronoi boundary. Figure 6 shows the template patterns for various critical points in two datasets. The red lines show the directions along which the seeds are placed in the saddle, center, and spiral templates. The bold red points represent the critical point locations and the blue points are the actual seed locations that form the seed template. Also note that if the size of the seed template was set to zero, then there will be no streamlines seeded based on flow information, and the resulting streamlines will all be seeded randomly with a Poisson disk distribution (see Section 3.4).

3.2 Controlling streamline density

The density of streamlines is determined to a large extent by the distance between the seeds. Once we have decided on the template patterns, it is straight forward to place seeds along these templates. We allow the user to control the density of the streamlines using a single parameter δ_{seed} which is the minimum distance between seeds. Seeds along the template are placed δ_{seed} apart from each other. Figure 6 shows the seeds along the templates for various critical points.

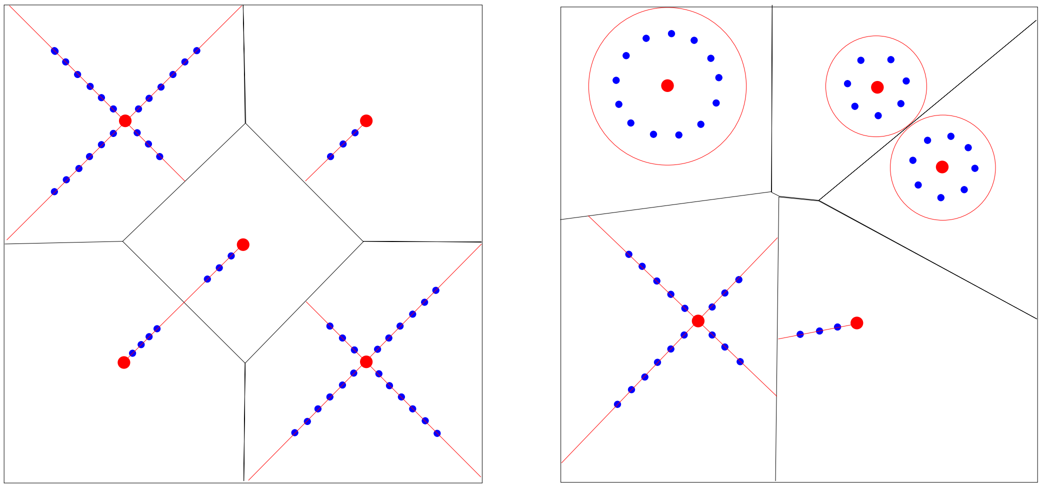

Figure 6: Seeds placed using templates for the various critical points. Left: dynamic vortices data set. Right: 5 critical points data set. These datasets are the same as those shown in Figure 2.

3.3 Terminating Streamlines

Our heuristic of seed placement according to templates works well to highlight the flow patterns around all the critical points present in the flow. If the streamline lengths are not chosen properly, then in some regions of the flow they will come too close to each other and create distracting clutter. We terminate a streamline when it comes a user specified distance close to an existing streamline. Let the user defined minimum separating distance desired between any two streamlines be $\delta_{streamline}$. The streamlines are represented as control points of a line strip. To check whether a given streamline is close to an existing streamline we determine whether the next control point during a streamline construction comes closer than $\delta_{streamline}$ to any control point of an already existing streamline. If the next control point of the current streamline is closer than $\delta_{streamline}$ to another control point of previously traced streamlines, then the current streamline is terminated.

To efficiently implement this streamline proximity test, we superimpose a cartesian grid over the flow field. Each cell of this grid contains a list of pointers to sample points of streamlines that fall in that cell. The width and height of each cell is equal to the desired separation $\delta_{streamline}$ between the streamlines. To check whether a sample point on a streamline is close to an existing streamline, all we need to do is to find the cell in which the sample point lies, and check whether it comes within distance $\delta_{streamline}$ of the points stored in the cell's eight neighbors. If the current control point is closer than $\delta_{streamline}$ from the points stored in cell's eight neighbors, then the streamline is terminated. For this test to be valid, it is required that the distance between successive sample points along a streamline be closer than $\delta_{streamline}$. We have found that in practice a value of $0.3 \times \delta_{seed}$ for $\delta_{streamline}$ works well.

During our investigation of streamline placement, we have experimented with several other streamline termination criteria but found the strategy to terminate a streamline based on its proximity to other streamlines to work the best. We have tried terminating streamlines based on a winding angle test to prevent clutter around a spiral or center type of critical point. We have also experimented with terminating streamlines when they exit the Voronoi region from which they were initiated, but found that the streamlines create distracting artifacts when any are terminated very close to straight edges of the Voronoi regions. Although the tests based on winding angle and crossing of Voronoi edges work well to reduce clutter, they do not generate evenly spaced streamlines.

3.4 Random Seeds

The placement of seeds according to templates for various critical points basically captures the location and behavior of the most interesting features of the flow. However, the strategy outlined above can leave “blank” spaces in our visualization where no streamlines are displayed. The flow in these regions is almost uniform and more or less parallel to their boundary. Figure 8 shows an example of streamlines generated using only the seed templates. The flow in these blank regions does not contain any additional features, hence we can afford to be less careful about the seed placement in these regions. Based on the observation that if the flow is uniform then we can place seeds arbitrarily as long as the streamlines do not crowd together to form distracting clusters, we chose to distribute seeds in these blank regions using a Poisson disk distribution [4]. Poisson disk distribution guarantees that these random seeds will be at least $\delta_{poisson}$ distance apart, where $\delta_{poisson}$ is chosen to be equal to δ_{seed}. Choosing $\delta_{poisson}$ to be equal to δ_{seed} ensures that all the streamlines will be evenly distributed.

We do not actually find these blank regions. Instead, we define a region of influence for all critical points and place random seeds according to a Poisson disk distribution outside the region of influence. The region of influence of a critical point is defined as a circle around the critical point location. The radius of this circle is decided based on the type of critical point and the size of its template seeding pattern. We choose the radius of the circle of influence to be a fraction of the size of the seed template. We found that the value of 0.8 works well. The yellow disks in Figure 7 show the regions of influence for each critical point.

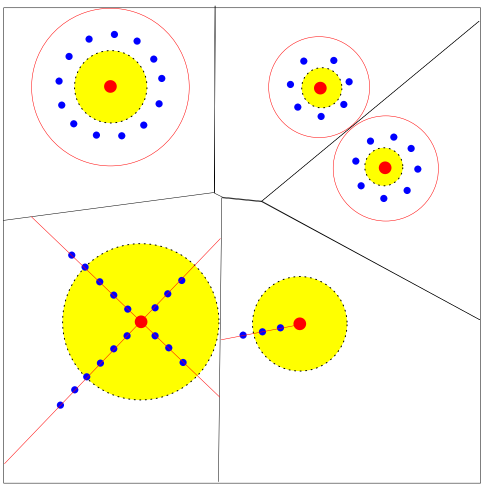

Figure 7: Regions of influence for each critical point are shown as yellow disks. The data set is the same as the one shown in Figure 1.

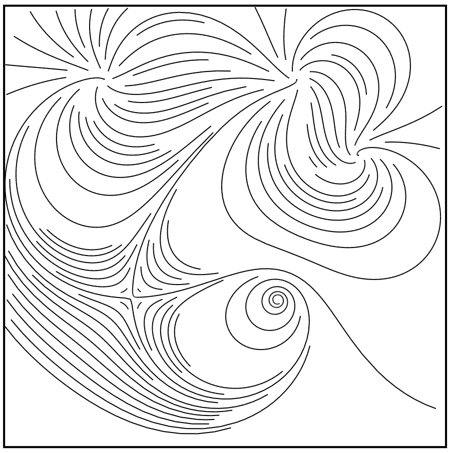

Figure 8: Streamlines generated using only the seed templates. The interesting features are captured by the streamlines but there are some blanks left that need to be filled by adding more streamlines.

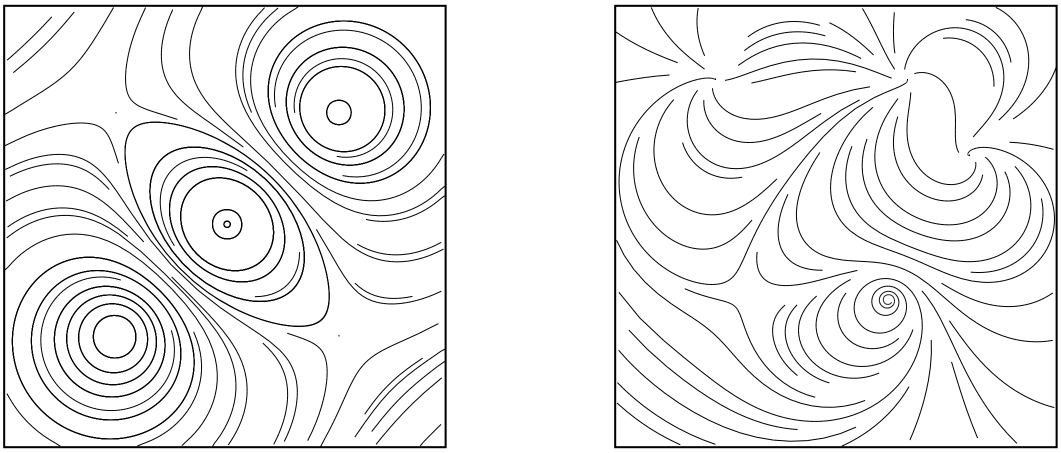

Figure 9: These images were rendered by using seeds whose locations were generated randomly using a Poisson disk distribution.

4 RESULTS

The first comparison is whether the flow-guided approach is any better than regular seeding or random seeding. For this, we placed seeds according to Poisson disk distribution and traced streamlines that were terminated using the proximity test. We found that while the strategy of placing seeds according to Poisson disk distribution works better than placing seeds on a regular grid, it still generates streamlines that do not sufficiently capture the flow close to critical points (especially saddles). As demonstrated in Figure 9, there are both aesthetic problems as well as missing critical flow information with random seeding strategy.

The next comparison is on how the flow-guided approach fared against the image-guided approaches. Figure 10 shows streamlines generated using our method for different values of δ_{seed} and $\delta_{streamline}$. These images show the critical points clearly at different streamline densities. The output using Turk and Banks's image-guided streamline approach is also presented side by side for

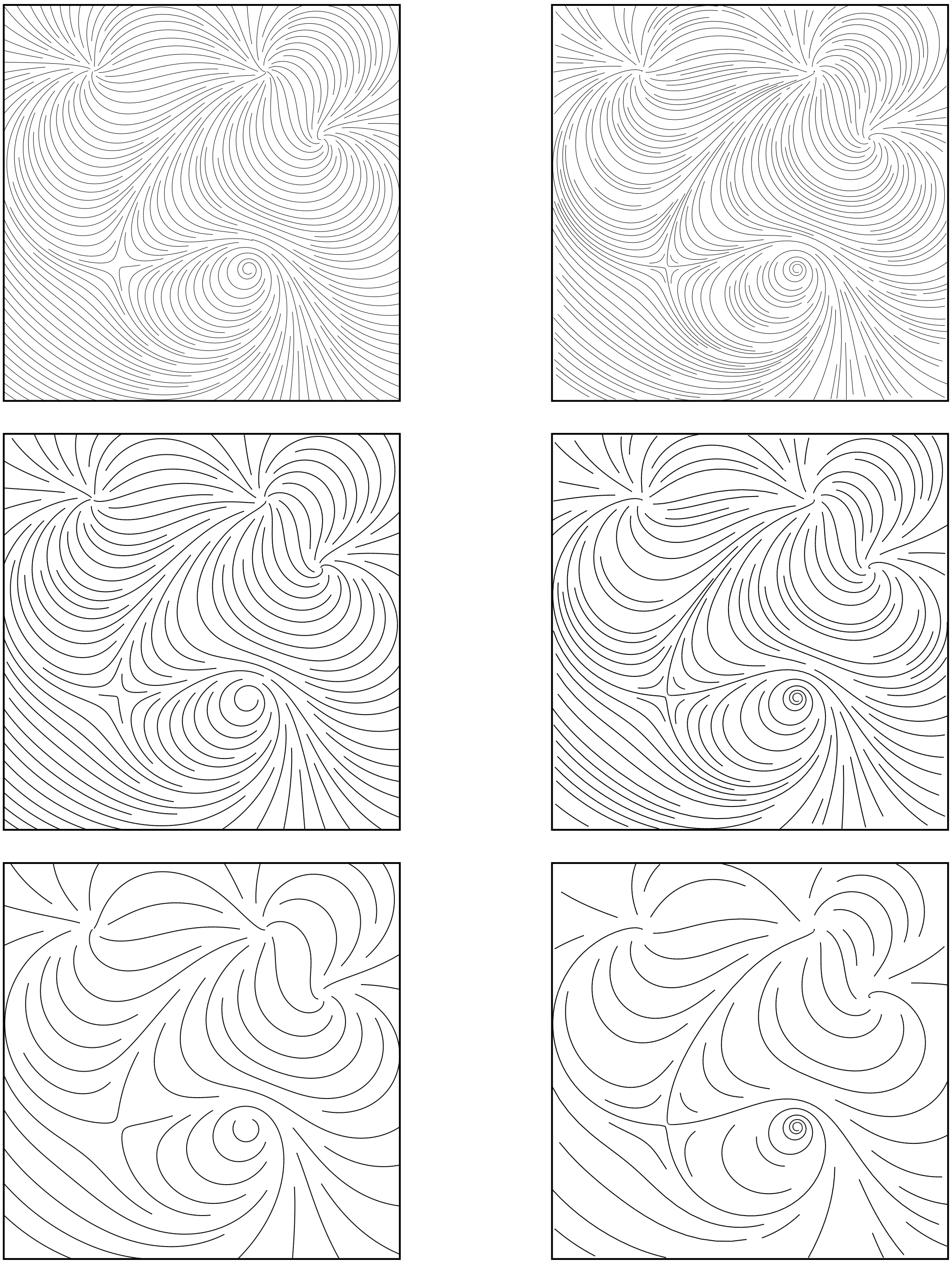

Figure 10: Left: images produced using Turk and Banks's image-guided streamline placement method. Right: our flow-guided streamline seeding method. The streamlines separation is chosen to be 1% of image width for images in the first row, 1.67% for the images in the second row, and 3% for images in the third row.

comparison. Efforts are made to bring the number of streamlines for corresponding densities as close as possible. The streamline separating distance $\delta_{streamline}$ was chosen to be $0.3 \times \delta_{seed}$. The values of δ_{seed} and $\delta_{streamline}$ which controls the density of streamlines roughly corresponds to the separating distance for Turk and Bank's method. On the top row, where the number of streamlines is very dense, the flow-guided approach captures the critical points better than the image-guided approach. This is the case even as we decrease the number of streamlines on the bottom row – particularly for saddles. The aesthetic quality, specially the uniformity of how the streamlines are distributed, seem to be better when using the image-guided approach as the number of streamlines is increased. This is attributed primarily to the effects of random Poisson disk streamlines. Note however as the number of streamlines increase, the computation cost of the image-guided streamline placement increases significantly.

Our observation is that streamline images generated using our flow-guided approach are comparable in quality to those produced using image-guided method of Turk and Banks [16] and the method of Jobard and Lefer [10]. Furthermore, our strategy guarantees that all the critical points present in the flow are highlighted by streamlines. Finally, the computational cost for seed placement is less using our method, specially as the number of streamlines increase.

5 FUTURE WORK AND SUMMARY

We have presented a technique for flow-guided seeding strategy that ensures coverage of important flow features, produce streamlines that are more or less uniformly spaced, and a greedy approach to producing long streamlines. The advantage over image-guided placement strategy is most obvious as the number of streamlines decrease. There are a number of extensions, improvements, and applications of this work that we are pursuing.

One of the motivations behind our research on streamline placement is to use as few streamlines as possible to represent the flow such that no important flow features are missed. Streamlines generated by our method can be used in texture synthesis techniques like PLIC [17]. PLIC uses texture mapped streamlines to generate visualizations that look like LIC images. However, due to the regular placement of the streamlines, the visualizations can have some distracting Moiré patterns. Generating streamlines using our method will eliminate the artifacts from PLIC images.

Streamlines are hard to use for studying 3D flow patterns because in three dimensions it becomes difficult to perceive the flow even if a moderate number of streamlines are used. The usual method is to use a rake to trace streamlines. While the rake approach is useful because the visualization is not very cluttered, the exploration of the flow using rakes is still quite ad hoc. The ideas in this work can be carried to three dimensions by identifying the critical points, partitioning the flow, and using suitable seed templates for three dimensional critical points. We also believe that a similar strategy to place streamlines in 3D will pave way for extending PLIC to 3D as well.

We have also observed that saddles can be better seeded if the seeds are placed along the bisectors of the topology lines of the saddle. This would increase the quality of seeding in the vicinity of the saddles. In addition, this would help deal with saddles with indices other than zero. Sometimes the distinction between sources and attracting spirals, and sinks and repelling spirals may become obscure using the parameterization reported in [2]. In this case, additional checks are necessary to determine which type of seed template is most appropriate.

Multiple levels-of-details, either in a zooming in/out operation or as required by a some rendering quota, can be easily supported by using seed templates with higher seed density δ_{seed}, and smaller streamline separation distance $\delta_{streamline}$. The idea is that only some of the seeds are traced if the viewpoint is far away (or the average number of streamlines per unit area of the screen is low). As the user zooms in for a closer inspection, more detail of the flow are presented by tracing out the other seed points and ramping up their alpha values to minimize popping.

ACKNOWLEDGMENTS

We would like to thank Chris Henze for his Zplane flow data generator, Jonathan Shewchuk for making the code for computing Voronoi diagrams publicly available, and Greg Turk and David Banks for their image-guided streamline placement code so that we can run some comparisons. We would also like to thank the members of the Advanced Visualization and Interactive Systems laboratory at UC, Santa Cruz for their feedback and suggestions. This project is supported by NASA grant NCC2-5281, LLNL Agreement No. B347879 under DOE Contract No. W-7405-ENG-48, DARPA grant N66001-97-8900, and NSF grant ACI-9619020.

References

[1] G. V. Bancroft, F. J. Merritt, T. C. Plessel, P. G. Kelaita, R. K. McCabe, and Al Globus. FAST: A multi-processed environment for visualization of computational fluid dynamics. In *Visualization '90*, pages 14–27, San Francisco, CA., October 1990. IEEE.

[2] Rajesh Bastra and Lambertus Hesselink. Feature comparisons of 3D vector fields using earth mover's distance. In *vis99*, pages 105–114, October 1999.

[3] M. S. Chong, A. E. Perry, and B. J. Cantwell. A general classification of three-dimensional flow fields. *Physics of Fluids A (Fluid Dynamics)*, 2(5):765 – 777, May 1990.

[4] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics*, 5(1), January 1986.

[5] Don Dovey. Vector plots for irregular grids. In *Proceedings: Visualization '95*, pages 248 – 253. IEEE Computer Society, 1995.

[6] Komei Fukuda. www.ifor.math.ethz.ch/ifor/staff/fukuda/polyfaq/node14.html.

[7] A. Gloubs, C. Levit, and T. Lasinski. A tool for visualizaing the topology of three-dimensional vector fields. In *Proceedings of Visualization '91*, pages 33 – 40, October 1991.

[8] J. L. Helman and Lambertus Hesselink. Surface represenations of two and three-dimensional fluid flow topology. In *Proceedings: Visualization '90*, pages 6 – 13. IEEE Computer Society, 1990.

[9] J. L. Helman and Lambertus Hesselink. Visualization of vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.

[10] B. Jobard and W. Lefer. The motion map: Efficient computation of steady flow animations. In *Proceedings of Visualization 97*, pages 323 – 328. IEEE, October 1997.

[11] Bruno Jobard and Wilfrid Lefer. Creating evenly-spaced streamlines of arbitrary density. In W. Lefer and M. Grave, editors, *Visualization in Scientific Computing '97*, pages 43 – 55. Springer, 1997.

[12] D.N. Kenwright, C. Henze, and C. Levit. Feature extraction of separation and attachment lines. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):135 – 144, April – June 1999.

[13] X. Mao, Y. Hatanaka, H. Higashida, and A. Imamiya. Image-guided streamline placement on curvilinear grid surfaces. In *Proceedings of Visualization '98*, pages 135 – 142, October 1998.

[14] N. Max, R. Crawfis, and C. Grant. Visualizing 3D velocity fields near contour surfaces. In *Proceedings of Visualization '94*, pages 248 – 255, October 1994.

[15] Jonathan Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. www.cs.cmu.edu/afs/cs/project/quake/public/www/triangle.html.

[16] Greg Turk and David Banks. Image-guided streamline placement. In *Proceedings SIGGRAPH*, pages 453–460, New Orleans, LA, August 1996. ACM SIGGRAPH.

[17] Vivek Verma, David Kao, and Alex Pang. PLIC: Bridging the gap between streamlines and LIC. In *Proceedings of Visualization '99*, pages 341 – 348, October 1999.