

Topology-driven Streamline Seeding for 2D Vector Field Visualization

Wenyao Zhang

School of Computer Science and Technology
Beijing Institute of Technology
Beijing 100081, China

Jianquan Deng

School of Computer Science and Technology
Beijing University of Posts and Telecommunications
Beijing 100876, China

Abstract—This paper presents a novel method of streamline placement for 2D vector field. In this method, the topological skeleton of underlying field is firstly extracted and used as initial streamlines. Initial streamlines segment the field into topological areas. Additional streamlines are then seeded at the center of topological areas in a recursive way, until there is no any valid empty area. To implement this method efficiently, a boundary extending of vector field is used to simplify the extraction of topology. And a virtual control grid superposed on the field is used to model topological areas approximately, and control the density of streamlines as well as their length. Our method focuses on the topology of vector field while keeping streamlines evenly-spaced as possible. But it can still run without the initial topological skeleton. Test results show that our method can achieve high quality of streamline placement.

Keywords—streamline placement, seeding strategy, boundary extending of vector field, vector field topology, virtual control grid, vector field visualization

I. INTRODUCTION

Streamline is a popular choice of vector field visualization for its fast and straightforward way to investigate flow patterns. As a sparse representation of vector field, the placement of streamlines directly determines the visualization quality. There must be a right number of streamlines at the right places. If there are too many, visual cluttering occurs in the scene. If too little, it is hard to form valid patterns.

To achieve effective placement, several streamline seeding methods have been developed, such as the image-guided [1], flow-guided [2], similarity-guided [3], and so on [4-8]. Some of these methods are based on the control of distance between streamlines, while the others use features of flow to guide the placement. In general, the distance-based methods can obtain evenly-spaced streamlines, but the cost of computation is rather high due to their blind or greedy search strategies. The feature-guided methods do placement in a more intelligent way, but they are not always available. In some special cases the features may be hard to detect, or does not exist at all.

With two-dimensional vector field in consideration, we present a novel streamline seeding strategy in this paper, which consists of two stages. The first is the extraction of topological skeleton. Topological skeleton will segment the vector field into regions having similar flow behavior. Such regions are called topological areas. Topological skeleton is then used as

initial streamlines. The second stage is to fill topological areas with additional streamlines. This is implemented by a recursive seeding at the centers of topological areas. To accelerate the placement, two special techniques are used to reduce the complexity and computation. One is the boundary extending of vector field that helps to detect topological skeleton. The other is a virtual control grid superposed on the domain of vector field. This grid is used to model topological areas and control the density of streamlines. With these techniques, we can do the placement of streamlines in a more efficient way.

The remainder of the paper is organized as follows. Section II reviews some existing algorithms in brief. Section III presents our placement method in detail. Test results and some remarks are given in Section IV. The final is the conclusions of this paper.

II. RELATED WORK

Many strategies have been proposed for streamline seeding in visualization literature. Most of them fall into two categories. One is to achieve evenly-spaced streamlines. The other is to highlight features of flow field. Here, only some related to two-dimensional vector field are outlined.

To obtain the specified density of streamlines, Turk and Banks [1] proposed an image-guided placement method. In their method, a flow field image is initialized by a set of randomly selected streamlines. And an energy function is used to measure the difference between a low-pass filtered image of streamlines and the desired uniform grey one. The placement is iteratively optimized to minimize the energy by some primitive streamline operations, such as moving, inserting, deleting, and so on. This early work can produce high quality placement, but its application is limited for its significant computation time.

Another method for creating evenly-spaced streamlines has been proposed by Jobard and Lefer [4]. In this method, a threshold for separating streamlines is preset, and an initial seed is put into a candidate queue. New streamlines are computed using seeds popped up from the candidate queue. During the computation, the distance of current streamline from all existing ones is always checked. If the current streamline has enough length, it will be saved, and some new seeds at a threshold distance away from it are selected into the candidate queue. To accelerate the computation, a Cartesian grid superposed to the vector field is used to control the inter-sample distance. This method is much faster than the image-

guided method. Due to its seeding strategy, however, this method will produce some undesirable effects in the results. Large empty spaces may remain unfilled. And discontinuities of streamlines may be noticeable near critical points or other specific areas.

To achieve a placement with long and evenly-spaced streamlines, Mebarki et al. [5] presented a greedy seeding strategy by choosing seed points at the farthest away from all existing streamlines. The implementation of this strategy is based on the Delaunay triangulation. Due to its high cost of computation, this strategy is not suitable for large data set. Another problem is the discontinuity occurred when the density of streamlines increases.

In order to minimize discontinuities and obtain high-density streamlines, Liu et al. [6] proposed an advanced evenly-space streamline placement algorithm, where a technique of double-queueing is used to prioritize the seeds around critical points as well as those introduced by long streamlines, and an adaptive distance control based on local flow variance is exploited to reduce cavities in placement.

The above methods all try to seed streamlines evenly. None of them has considered the features of underlying flow.

With the goal to capture flow patterns near critical points, Verma et al. [2] presented a flow-guided streamline seeding method. In their method, critical points are first identified, and flow field is segmented into regions, each containing a single critical point. Then a seeding template determined by the type and location of critical point is applied to each region. Finally, some blanks left in the field are filled by additional seed points, which are randomly selected according to Poisson disk distribution. The main advantage of this method is the flow patterns around critical points are kept in the placement, even when the streamline density is low. The disadvantage is the uniformity of streamlines is not always satisfied. If there are not enough streamlines, the density disparity may be obvious. Moreover, this method is failed when no any critical point exists in the field.

In [7], a similar method taking flow features into account is presented, where periodic orbits and separatrices in the field are extracted as initial streamlines. This initialization makes the topology of vector field and the continuity of periodic orbits maintained in the placement. But this goal is achieved at the cost of a complex Morse decomposition for the field.

In [8], Li et al. proposed an illustrative placement method, which can highlight the essential flow features, and provide visual focuses, by placing more streamlines in the regions with low spatial coherence. Here, the spatial coherence is obtained by evaluating the dissimilarity between streamlines locally and globally. This method does not favor uniformly spaced streamlines. And its computation is also high due to the measurement of spatial coherence. It may be a disadvantage in some applications. Another method highlighting flow features is presented in [3]. Its main difference from Li et al. [8] is the selection and growth of streamlines, which are based on the local similarity between streamlines. The disadvantage of this method is still the cost and complexity of similarity computation that has to be done for each sample.

III. OUR METHOD

The main goal of our method is to highlight the structure of flow field in a placement while keeping the streamlines evenly-spaced as possible. In addition, we hope our algorithm run fast, so that it can be used to interactively analyze large scale of data set. To achieve the goals, we include a procedure of feature extraction in our method, by which a topological skeleton is obtained from flow field. The topological skeleton consists of critical points and limit streamlines. Both of them make a partition for the domain of vector field. And each part is a topologically uniform region, which is referred to as topological area in this paper.

In our placement, topological skeleton is used as initial streamlines. And a special technique of boundary extending for vector field is proposed to simplify the extraction of topological skeleton. In the following process, additional streamlines are placed with the help of a virtual control grid, which is superposed onto the vector field domain. The virtual control grid plays an important role in our algorithm. It is used to control the density of streamlines, and model the topological areas approximately. Moreover, it is the virtual control grid that accelerates the placement of streamlines. Except the initial streamlines, all other streamlines are seeded at the centers of empty topological areas. This central seeding strategy is reasonable in terms of topology. Because all streamlines within a topological area have the same flow behavior, i.e., all start from a source and end at a common sink. If only one streamline is permitted to link the source and the sink, the one through the center may be the best choice. In this sense, our seeding is a topology-driven strategy.

For a two-dimensional vector field defined on a rectangular grid, our placement algorithm works as followed:

- 1) Extend the vector field by mirroring boundary cells, as shown in Fig.1.
- 2) Define a virtual control grid on the original domain using a predefined cell size, and set all control cells to empty.
- 3) Do the standard topology analysis for the extended vector field to extract its topological skeleton.
- 4) Clip the topological skeleton, deleting or truncating all components out of the original domain.
- 5) Place the streamlines in the topological skeleton as initials, and set all control cells covered by initial streamlines to the number of streamlines having traveled through them.
- 6) Scan the virtual control grid, and find an empty topological area, i.e., a set of continuous empty cells.
- 7) Seed a streamline at the center of the empty topological area, and update the counters of control cells covered by the line. The streamline integration terminates at the reaching of critical point, going out of the field or entering a full control cell. This termination policy can ensure that the additional streamlines have a desired separating distance.
- 8) Repeat steps 6 and 7 until there is no any valid topological area that is model by empty cells in the virtual control grid.

Some concepts and processes involved in this algorithm are further addressed in the following subsections.

A. Boundary Extending of Vector Field

For a two-dimensional vector field $F:(x,y) \mapsto (u,v)$, defined on the domain $D=[x_{\min},x_{\max}] \times [y_{\min},y_{\max}]$, it is often represented by a discrete grid with rectangular cells $C_{i,j}$ ($i=1, \dots, M, j=1, \dots, N$). Each cell has a vector $(u_{i,j}, v_{i,j})$ defined on its center $(x_{i,j}, y_{i,j})$. Such a field is shown in Fig. 1, which is marked by real lines.

Before the analysis of topology, we extend the vector field by adding a virtual boundary around the domain, as shown in Fig.1, where the added cells are drawn in dashed lines. The virtual cells in the extended boundary have the same size as the real ones. By this means, the domain of vector field is also extended to a new one.

To define the vectors in virtual cells, we mark the newly added cells with subscripts 0, $M+1$ and $N+1$, and use them in the same way as real ones. The vectors for virtual cells are then set as expressed in (1) and (2):

$$\begin{cases} (u_{0,0}, v_{0,0}) & = (-u_{1,1}, -v_{1,1}) \\ (u_{0,N+1}, v_{0,N+1}) & = (-u_{1,N}, -v_{1,N}) \\ (u_{M+1,0}, v_{M+1,0}) & = (-u_{M,1}, -v_{M,1}) \\ (u_{M+1,N+1}, v_{M+1,N+1}) & = (-u_{M,N}, -v_{M,N}) \end{cases} \quad (1)$$

$$\begin{cases} (u_{0,j}, v_{0,j}) & = (u_{1,j}, -v_{1,j}) \quad j=1, \dots, N \\ (u_{M+1,j}, v_{M+1,j}) & = (u_{M,j}, -v_{M,j}) \quad j=1, \dots, N \\ (u_{i,0}, v_{i,0}) & = (-u_{i,1}, v_{i,1}) \quad i=1, \dots, M \\ (u_{i,N+1}, v_{i,N+1}) & = (-u_{i,N}, v_{i,N}) \quad i=1, \dots, M \end{cases} \quad (2)$$

Equation (1) is for the special cells in corners, and (2) is for the vertical and horizontal borders. In fact, this extending is a vector mirroring along the field borders, as demonstrated in Fig.1.

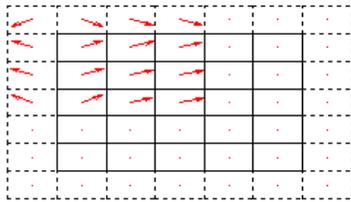


Figure 1. Demonstration for boundary extending of vector field.

B. Virtual Control Grid

Besides the grid of vector field definition, another grid is used in our algorithm. This grid covers the original vector field exactly, but the width and height of its cells are set to a predefined value. We call it virtual control grid. All control cells in this grid are set to empty at first. During the streamline placement, they will count the lines having traveled through them. A max number is preset for allowing a control cell to

hold streamlines. When it has the max number of streamlines, it becomes a full cell. The streamline integration will stop at the entering of any full cell. Therefore, the separating distance between streamlines is controlled at a desired level. Although this idea is similar to the cell-based distance control in [4], the efficiency is significantly improved by replacing the computing of distance with a counting of streamlines.

Another usage of control grid is to model the topological areas in our method. According to the definition, a topological area is a sub-region closed by streamlines or boundaries. In the virtual control grid, such a topological area corresponds to a set of continuous empty cells. And these cells describe the shape and size of the topological area approximately.

C. Topological Analysis

Topological skeleton of a vector field can be extracted using the standard method proposed by Helman and Hesselink in [9]. With this method, all critical points are detected and classified into sources, sinks, centers or saddle points. For each saddle point, along the directions of its eigenvectors, we can compute four special streamlines that start or end at the point. These special lines are in fact the limit streamlines of flow field. All of critical points and limit streamlines form the skeleton of topology. However, this concise extraction of topological skeleton is not always effective, because it can not detect the open separation and attachment lines, which are also limit streamlines but without any associated saddle point. For these open limit streamlines, a specific method has been proposed in [10]. But this method is not suitable for our purpose of streamline placement, because its adoption increases the complexity of computation.

Therefore, we take a special technique to deal with the open limit streamlines. We extend the vector field domain before topological analysis by adding a virtual boundary, as addressed in Section III-A. The extraction of topological features is performed on the extended field using standard method. The great benefit of boundary extending is that the special open limit streamlines can be processed in the same way as the ordinary closed ones that are associated with saddle points. This can be demonstrated by the following example.

Fig.2 (a) shows a flow field with streamlines, which is defined by (3):

$$\begin{cases} u = (x^2 + 2xy + y^2)/2 \\ v = (-1 + x^2 + 2xy - y^2)/2 \end{cases} \quad (3)$$

It can be deduced that there are two limit streamlines in this field, but no any critical point. We can't get these two lines by the means of critical points. After adding a virtual boundary, we can see that, from Fig.2 (b), some critical points are also added into the field. And there are two saddle points that are respectively linked to the two limit streamlines. Therefore, we can extract the limit streamlines directly through the newly added saddle points.

After deleting or truncating all topological components outside of the original domain, we get the final topological skeleton.

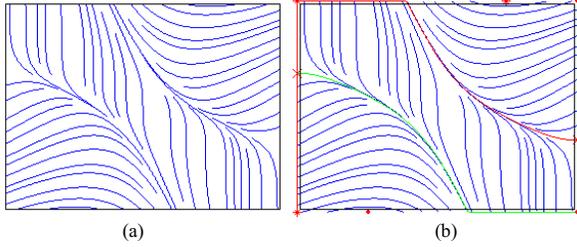


Figure 2. Example for topological analysis: (a) is a flow field with open separation and attachment lines; (b) shows the extended field with topological skeleton, where sources, sinks and saddle points are respectively marked by *, · and x.

D. Seeding in Topological Areas

Separatrices of topology are used as initial streamlines in our placement. Additional streamlines are added in a recursive way within topological areas. For an area modeled by control cells, we first compute its geometric center. If the center lies in the area, it is then selected as a seed for streamline placement. Otherwise, we draw two lines through the center vertically and horizontally. The topological area intercepts both of them. The midpoint of the longest segment is then selected as a seed of streamline. Fig. 3 shows a typical example of this case, where the midpoint S of segment AB is used for seeding.

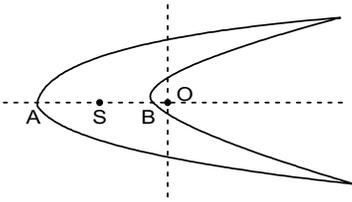


Figure 3. Seed selection for a concave area

Once the seed is determined, a streamline is placed at the point by a method of line integration. During the placement, the counters of control cells are updated to remember the number of streamlines having traveled through them. The integration stops at the reaching of critical point, field boundary or full control cell. The newly added streamline bisects the original area. Each part of them is still a topologically uniform area, where the seeding can be continued in the same way.

This seeding strategy in topological areas implicitly sets up a binary tree describing the logical relationship among streamlines, which is useful for building a hierarchy of multi-resolution placement of streamlines.

IV. TEST RESULTS

The quality of our streamline placement depends on two factors. The first one is the cell size of virtual control grid, which is denoted by the δ parameter. The second is the max number of streamlines that a control cell can hold, which is marked by the λ parameter. To favor long streamlines, we can set λ to a bigger value. Fig.4 shows examples of placement with different λ values for the flow field used in Fig.2. The visual effect of the right image in Fig.4 is better than the left one because of the prolonging of some lines. The parameter λ

also affects local density of streamlines as shown in Fig.4. The whole density of streamlines inversely varies with the value of δ . Such examples are shown in Fig.5. Therefore, the parameter δ must be set carefully, because too small value causes visual cluttering. Besides the above two factors, we can control the placement in another way by leaving the small empty topological areas unfilled. This can be easily incorporated into our algorithm.

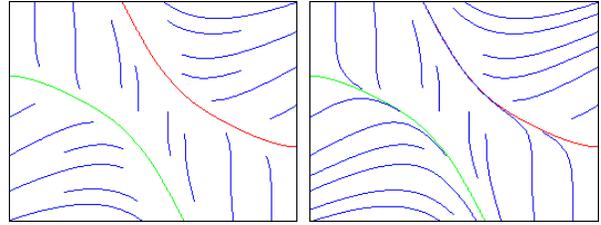


Figure 4. Placements with different λ values for the flow field used in Fig.2: $\lambda=1$ for the left image, and $\lambda=3$ for the right image.

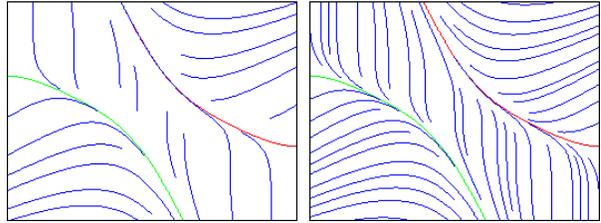


Figure 5. Placements with different δ values for the flow field used in Fig.2: in the left image, δ is 10% of the field width, and 5% for the right image.

To evaluate the quality, we compare our placements with those generated by the demo program [11] of Mebarki et al.'s algorithm under the condition of similar separating distance. Here, two typical comparisons are given in Fig.6. The first is for a vector field enclosed with the demo program. The placements of this field are shown in the first row of Fig.6, where the left image is output by the demo program, and the right is generated by our method. In the right image, the streamlines of topological skeleton are clearly presented as well as the critical points. In the left image, we have to infer the topological skeleton from the placed streamlines, most of which are broken into segments of short lines. The critical points, special for the saddle points, also fall into this case. Besides these topological features, the number of short streamlines in the left image is much more than the right one. All of these distinctions can be seen in another comparing test, which is given in the second row of Fig.6. The vector field for this test is generated by Clifford algebra according to the method in [12].

As shown in Fig.6, the placements produced by our method are better in quality than the ones generated by Mebarki et al.'s algorithm. Currently, the efficiency is not evaluated in quantity due to the means of implementation. But we believe that our method can run fast because of the techniques taken to reduce computation, as mentioned in Section III. It should be noted that, however, our placement is only uniform in approximation. Local density of streamlines may vary in different areas. But the density disparity is under the control of δ and λ .

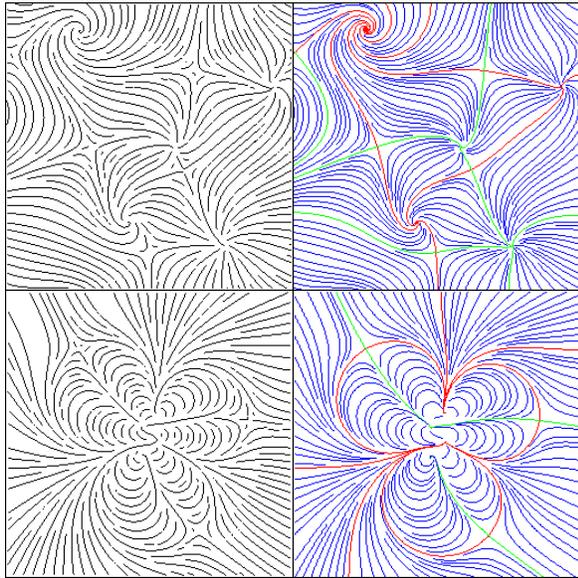


Figure 6. Comparative results of placements by different methods: both images in the left column are generated by the method of Mebarki et al., and the right ones are produced by our method.

Furthermore, if there are too many streamlines cluttered together, we can reduce the intensity of lines by a post processing of rendering the final image in alpha mode, just as done in [8]. Here the alpha value can be obtained from the counter value of each control cell. These are byproducts of our placement. Finally, contributing to the bisection of topological areas, our method can be used for multi-resolution visualization. We can adjust the placement in an easy way.

V. CONCLUSIONS

We present an intelligent method of streamline placement for 2D vector fields, which includes an extraction of topology and a recursive streamline seeding in topological areas. The extraction of topology leads to a clear flow structure in placement. The recursive seeding makes it suitable for multi-resolution visualization. Besides these, our method simplifies topological analysis by extending the boundary of vector field. And the placement is accelerated by the use of a virtual control

grid, which assists us to control the length and density of streamlines. Although the placement is not absolutely evenly-spaced, it is almost uniform in global view. The disparity of density can be adjusted by placement parameters. In addition, our method can still run without any information about topology, although it is a deterministic method driven by topology. Test results show that our method can achieve high quality of placement. In future work, we will optimize its implementation and evaluate its performance sufficiently.

REFERENCES

- [1] G. Turk and D. Banks, "Image-guided streamline placement," In Proc. SIGGRAPH '96, 1996, pp. 453–460.
- [2] V. Verma, D. Kao, and A. Pang, "A flow-guided streamline seeding strategy," In Proc. IEEE Visualization '00, 2000, pp. 163–170.
- [3] Y. Chen, J. D. Cohen, and J. H. Krolak, "Similarity-guided streamline placement with error evaluation," IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 6, pp. 1448–1455, 2007.
- [4] B. Jobard and W. Lefer, "Creating evenly-spaced streamlines of arbitrary density," In Proc. 8th EG Workshop on Visualization in Scientific Computing, 1997, vol. 7, pp.43–56.
- [5] A. Mebarki, P. Alliez, and O. Devillers, "Farthest point seeding for efficient placement of streamlines," In Proc. IEEE Visualization '05, 2005, pp. 479–486.
- [6] R. J. M. I. Zhanping Liu and J. Groner, "An advanced evenly-spaced streamline placement algorithm," IEEE Transactions on Visualization and Computer Graphics, vol.12, no.5, pp. 965–973, 2006.
- [7] G. Chen, K. Mischaikow, R.S. Laramée, P. Pilarczyk, and E. Zhang, "Vector field editing and periodic orbit extraction using Morse decomposition," IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 4, pp. 769–785, 2007.
- [8] L. Li, H.-H. Hsieh, and H.-W. Shen, "Illustrative streamline placement and visualization," In IEEE 2008 Pacific Visualization Symposium, 2008, pp. 79–86.
- [9] J. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets," IEEE Computer, vol. 22, no. 8, pp. 27–36, 1989.
- [10] D. N. Kenwright, "Automatic detection of open and closed separation and attachment lines," In Proc. IEEE Visualization '98, 1998, pp. 151–158.
- [11] A. Mebarki. Demo Executable. <http://www-sop.inria.fr/geometrica/team/Abdelkrim.Mebarki>.
- [12] G. Scheuermann, H. Hagen, and H. Krüger, "An interesting class of polynomial vector fields," In Mathematical Methods for Curves and Surfaces II, M. Dæhlen, T. Lyche, and L. L. Schumaker, Eds. Nashville: Vanderbilt University Press, 1998, pp. 429–436.