Difference-Contribution Strategy for Seeding 2D Streamlines

Shaorong Wang, Rui You, Yisong Chen, Sheng Li, Guoping Wang

The Key Lab of Machine perception and intelligent, MOE,Beijing, China, 100871

School of Electronics Engineering and Computer Science, Peking University, Beijing, China, 100871

{wangsr@graphics.pku.edu.cn, yourui@graphics.pku.edu.cn, chenys@graphics.pku.edu.cn, lisheng@graphics.pku.edu.cn, gwang@graphics.pku.edu.cn}

ABSTRACT

This paper presents a novel seeding strategy for streamline visualization of 2D vector field. The main idea of our approach is to capture the spatial-varying features in a vector field. Generally speaking, we measure the difference between the inflow and the outflow to evaluate the local spatial-varying feature at a specified field point. A Difference-Contribution Matrix (DCM) is then calculated to describe the global appearance of the field. We draw streamlines by choosing the local extreme points in DCM as seeds. DCM is physics-related thus reflects intrinsic characteristics of the vector field. The strategy performs well in revealing features of the vector field even with relatively few streamlines.

Keywords

Seeding strategy, Streamline, Difference-Contribution Matrix

1. INTRODUCTION

Vector fields are commonly used in many scientific and engineering domains, such as astronomy, aeronautics, and meteorology. Visualization of vector fields is important for properties analysis. The most common approaches include geometry-based, texture-based, feature-based, and streamline-based approaches.

Geometry-based approaches, such as arrow and hedgehog plots, give a visual perception of local flow feature.

Texture-based methods give a dense representation of the vector field. However, they can't provide visual focuses on significant information of vector field and obtain visually pleasing images requires an intrinsically huge computational expense.

Feature-based visualization approaches seek to compute a more abstract representation that already contains the important properties in a condensed form and suppresses superfluous information. Anyway, the feature is always not easy to be extracted.

The most popular flow visualization method in use today is still streamlines and those derived from streamlines because they provide sparse visualization that focus on significant structures and can be combined with other visualization techniques. Furthermore, they are faster to compute and can be rendered at any resolution at interactive rates.

The quality of visualization of the streamlines highly relies on the seeding strategy, which includes seed location and a length of each streamline. In other words, it's very important to select a set of streamlines to represent the vector fields comprehensibly and completely. On the one hand, placing too many streamlines can make the final images cluttered, and hence make it more difficult to understand the data. On the other hand, we may miss important flow features if too few streamlines placed. An ideal streamline seed placement algorithm should be able to generate visually pleasing and technically illustrative images.

There are several seeding strategies developed in the past years, such as evenly-spaced streamlines algorithm [Liu06], and feature-guided algorithm. A criteria of seeding strategy is proposed by Verma et al. [Ver00]. Coverage, no important features of the vector field should be missed and the streamlines should cover the whole domain; Uniformity, the distribution of streamlines should be more or less uniform across the domain; Continuity, long continuous streamlines are preferred over short ones.

In this paper, we define a Difference-Contribution Matrix (DCM) as a metric for flow features. We propose a novel 2D streamline seeding strategy according to the DCM. Suppose a region including inflow and outflow shown in Figure 1, there is cross interface between the flow and the region. If the area of inflow interface is not the same as that of outflow interface, changes happen in the region. The greater the difference between the inflow and the outflow, the greater the vector fields change.

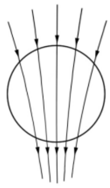


Figure 1. Inflow and outflow

Compared to the past approaches, the strategy proposed in this paper give higher priority to the variation of the streamline than to the density of the streamlines. This is because the former represents more flow feature. In other words, if there is little variation in a region, the streamline is nearly evenly distributed in the region and they can be represented by fewer streamlines. If the variation is great in a region, more streamlines are needed to provide the detail.

The seeding strategy in this paper is based on the DCM. Streamline starting points are seeded depending on the maxima of the matrix. Because DCM is defined by the physical meaning of the vector fields, our seeding strategy is able to qualitatively capture more important flow features with less streamlines, hence less clutter and occlusion.

The advection of streamlines in the previous streamline placement algorithms can be terminated by explicit inter-streamline distance control. This may cause visual discontinuity of the flow pattern, especially when it is near the vicinity of critical points. Our seeding algorithm only determines complete streamlines which are integrated as long as possible until they leave the domain, reach a critical point, or generate a loop. Without abruptly stopping the streamlines, the flow patterns shown in the visualization are much easier to understand.

2. RELATED WORK

Overview of vector field visualization techniques can be found in [Lar04] and [Pos03]. We consider here the most relevant work in streamline visualization. A number of techniques with different objectives have been developed. We group the present seeding strategies into four categories: image based, direct, feature based and vector field property-based.

Image-based method searches for an energy function's minimal value to place seeds, in which the energy function is defined in image space according to streamlines. In [Tur96], techniques for automated placing of seed points were developed to achieve a nearly uniform, dense distribution of streamlines for 2D flow fields. Mao et al. [Mao98] extend this approach to 3D curved surfaces. For 3D flow fields, seeding strategies typically involve analysis of the underlying flow field to visualize certain features using sparse distributions.

Direct methods place new streamlines with a certain heuristic rule without computing any global energy function. A seeding strategy for automated placing of seed points was developed to achieve a nearly uniform, dense distribution of streamlines for 2D flow field [Job97]. The technique is extended to unsteady flows in [Job00], and multi resolution flow visualization in [Job01]. By defining a 3D Euclidean distance metric, the strategy is directly extended to 3D field [Mat03]. The seeding strategy presented by Mebarki et al. [Meb05] starts new streamlines in the center of the biggest remaining voids, and achieve good continuity and uniformity of the streamlines by a greedy algorithm. Liu et al. [Liu06] improves continuity by prioritizing streamline elongation over new streamline insertion.

Feature-based flow visualization is concerned with the extraction of specific patterns of interest, or features. Verma et al. [Ver00] first proposed a feature-based strategy for 2D vector field visualization. The seeding strategy is extended to 3D vector fields by Ye et al. [Ye05].

Streamline similarity and streamlines density are both properties of vector field. They can be regarded as the criteria of adding new streamlines. Li et al. [Li07] proposed a 3D image-space streamline placement method. They control the seeding and generation of streamlines in image space to avoid visual cluttering. Schlemmer et al. [Sch07] defined the streamline density of a region as the ratio between the number of occupied pixels by streamlines and the total number of pixels in the region.

3. DISTRIBUTION-BASED SEEDING STRATEGY

3.1. In-out Contribution Matrix

We first give some definitions about our idea. For a non-zero vector at any position in a vector field, there is a streamline passing through the position. A streamline is a Complete Streamline if either of the following conditions is satisfied:

The ending point overlaps the starting point. In other words, the streamline is a closed curve.

The endpoint is on the border of the vector field, or the vector at the endpoint is zero.

First a set of Complete Streamlines are generated to cover the vector field domain, which is called as the Complete Streamline Set. The Complete Streamline Set can be generated uniformly or randomly. The former method is chosen in this paper: The vector field domain is evenly divided into $m \times n$ squares, and then streamlines are seeded at each square's center. If all the streamlines are regarded to be different, we get a Complete Streamline Set with $m \times n$ Complete Streamlines.

For a given point p in the vector field, c_p is a circle of radius r centered at p. We partition the circle c_p into congruent curve segment units uniformly. Each unit u_i has an outward-weight $w_{out}(u_i)$ and an inward-weight $w_{in}(u_i)$, both of which are initialized with 0 and $0 < w_{in}, w_{out} < 1$. Given a Complete Streamline Set S_{line}, subset S_{sub} contains all streamlines in S_{line} which have intersection with c_p. For each streamline l in S_{sub}, cp is the intersection point of l and c_p, $N(p)$ is the number of all intersection points of subset S_{sub} and c_p. Let $\mathbf{V}$ be the vector at the intersection point cp, if $\mathbf{V}$ is outward to the circle c_p, cp is called as an outward intersection point, otherwise it is an inward intersection point. For every inward intersection point cp_i, we calculate its inward contribution $\mathrm{Con}_{\mathrm{in}}(cp_i, u_j)$ to every unit u_j:

$$\mathrm{Con}_{in}(cp_i, u_j) = \mathrm{F}(Dis(cp_i, u_j))$$

$\mathrm{Con}_{in}(cp_i, u_j) = \mathrm{F}(Dis(cp_i, u_j))$ Where $Dis(cp_i, u_j)$ is the distance between cp_i and u_j, and $\mathrm{F}(\ )$ is a decreasing function.

The weight of every unit u_j is updated by every inward intersection point cp_i:

$$w_{in}(u_j) = w_{in}(u_j) + Con_{in}(cp_i, u_j)$$
$$w_{in}(u_j) = 1, \text{ if } (w_{in}(u_j) > 1)$$

The inward-contribution of point   is defined as

$$\mathrm{Con}_{in}(p) = \sum_j w(u_j)$$

And the outward-contribution is calculated the same as that of inward-contribution.

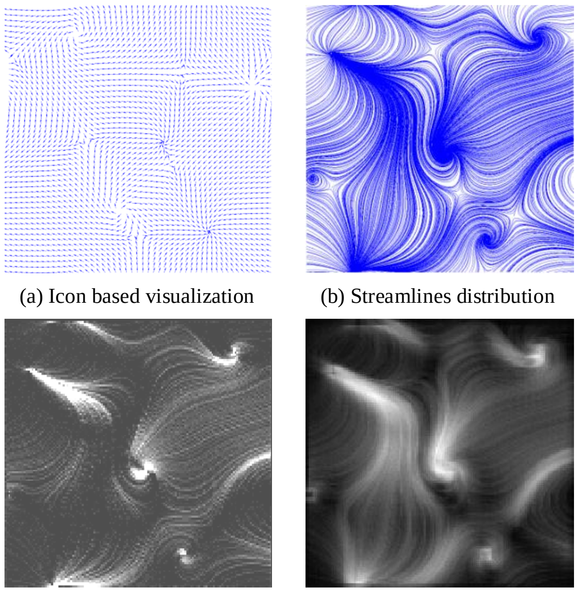
Support points have been sampled uniformly in the vector field, for each sampling point $p(i, j)$ we calculate its inward and outward contribution $\mathrm{Con}_{in/out}(p)$. Then the Density Matrix $\mathbf{Mat}_{density}$, Out-Contribution Matrix $\mathbf{Mat}_{out}$, In-Contribution Matrix $\mathbf{Mat}_{in}$, Signed-Difference-Contribution Matrix $\mathbf{Mat}_{sdelta}$ and Difference-Contribution Matrix(DCM) $\mathbf{Mat}_{del}$ can be defined as:

$$\mathbf{Mat}_{density} = (\mathrm{N}(p(i,j)))$$
$$\mathbf{Mat}_{out} = (\mathrm{Con}_{out}(p(i,j)))$$
$$.\ \mathbf{Mat}_{in} = (\mathrm{Con}_{in}(p(i,j)))$$
$$\mathbf{Mat}_{sdelta} = \mathbf{Mat}_{in} - \mathbf{Mat}_{out}$$
$$\mathbf{Mat}_{del} = abs(\mathbf{Mat}_{sdelta})$$

The following statements of DCM are obvious:

1. For any element a in $\mathbf{Mat}_{in}$, $\mathbf{Mat}_{out}$, $a > 0$
2. If $\mathbf{Mat}_{in}(i,j) > 0$ and $\mathbf{Mat}_{out}(i,j) = 0$, there exists convergent points around $p(i,j)$.
3. If $\mathbf{Mat}_{in}(i,j) = 0, \mathbf{Mat}_{out}(i,j) > 0$, there exists divergent points around $p(i,j)$.
4. If $\mathbf{Mat}_{sdelta}(i,j) > 0$, $\mathbf{Mat}_{in}(i,j) > \mathbf{Mat}_{out}(i,j)$, a flow will be "squeezed" when the flow p passes through the region around $P(i,j)$.
5. If $\mathbf{Mat}_{sdelta}(i,j) < 0$, $\mathbf{Mat}_{in}(i,j) < \mathbf{Mat}_{out}(i,j)$, a flow will be "expanded" when the flow passes through the region around $p(i,j)$.

From above definition, DCM is somewhat like divergence. The divergence represents the volume density of the outward flux of a vector field from an infinitesimal volume around a given point. The divergence of the velocity field in that region would have a nonzero value only when the region is a source or sink. As shown in Figure 1, if there is no sink or source in the region, divergence is 0. On the contrary, the length variation between inflow interface and outflow interface is nozero, which is



(a) Icon based visualization (b) Streamlines distribution

(c) DCM (d) Density matrix

Figure 2. Vector field and its statistics matrix

described by our DCM.

Figure 2(a) shows icon based visualization result. Figure 2(b) shows the sample streamlines. Figure 2(c) shows DCM and Figure 2(d) shows the density matrix. Vector field variation is more enhanced in DCM than that in Density Matrix. The density of the consistent region may be very higher, while the value of DCM may be very little.

3.2. DCM seeding strategy

We try to sort the seeds according to the variation of the vector field. A seed with greater variation has higher priority.

In this section, DCM defined in the past section is used to represent the variation the vector field. According to this DCM streamline start points are seeded mainly depending on the maxima of the matrix. The generation of each streamline lowers the matrix locally until the given condition is satisfied.

3.2.1. Initialization

To start our iterative seeding strategy, we need an initialization set of streamlines. The maxima of DCM can be regarded as the initial seed. As the streamlines vary greatly around the elements of big values in the DCM, and the feature are more evident. If there are several candidate seeds with the same value, we randomly get one from the candidates. Thus if we assume a constant DCM, start points are generated randomly and would not be picked in a raw.

If there are some critical points in the vector field, the topology structure is an import property of the vector field. To discover the vector field's detail, seeds around the critical points are preferred. DCM captures sources or sinks nodes easily. On the other hand, streamline around a saddle are much less than around other positions. So seeds around saddle are placed firstly. The location and classification methods of critical points can be found in [Gre92] and [Hel89] [Hel91].

3.2.2 Iteration

Each of the iteration consists of two major parts:

1. Trace a new complete streamline in forward and backward direction and test for intersections.
2. Update the DCM according to the new streamline.

In step 1, new seed is picked by get the maxima of DCM. As described in the initial step, if there are several candidate seeds with the same value, we randomly get one from the candidates.

The element priority of DCM around the new streamline is lowered after the streamline is added. If the DCM is not updated, the next candidate seed may be very close to the previous one and the generated streamlines are also very close to each other. So an update process is taken after a new complete streamline is added.

Obviously the influence from the new streamline on the vector field's feature of a given region is related to the distance between the streamline and the region. For a given new streamline, we first get all streamlines' positions in DCM, which is denoted as a position set S_p. All the elements of these positions are set to 0, which means that no streamline will be added more than once. The other elements in DCM are updated by their distances to the set S_p. For a given position p, the value $DCM(p)$ is updated by a function $\mathrm{F}_{update}()$ as follows:

$$DCM(p) = \mathrm{F}_{update}(Dis, DCM(p))$$

Where Dis is the distance between p and set S_p. For a given $DCM(p)$, Dis is non-negative. The longer Dis is, the smaller $DCM(p)$ is. In other words, the farther away from the region, the less influence the new streamline has on the region.

If the distances between all position and the set SP are calculated during DCM update process, too many CPU resources will be consumed. Given a maximal distance $d_{\max}$, if we have $d > d_{\max}$, then $DCM(p)$ is the same as the previous value. So we only update those values whose distances to set SP are no more than $d_{\max}$. Inspired by [Set99], a fast marching method is adopted in this paper.

If seeds around the saddles are placed firstly, we update the DCM when all the streamlines from the saddle seeds are generated.

3.2.3. Termination

The algorithm terminates if either of the following happens.

- The number of streamlines is greater than a given value. If the number is too small, some important detail may be missed.
- DCM satisfies some conditions, such as the minimum of DCM is smaller than the given value, which means the most important feather is captured.

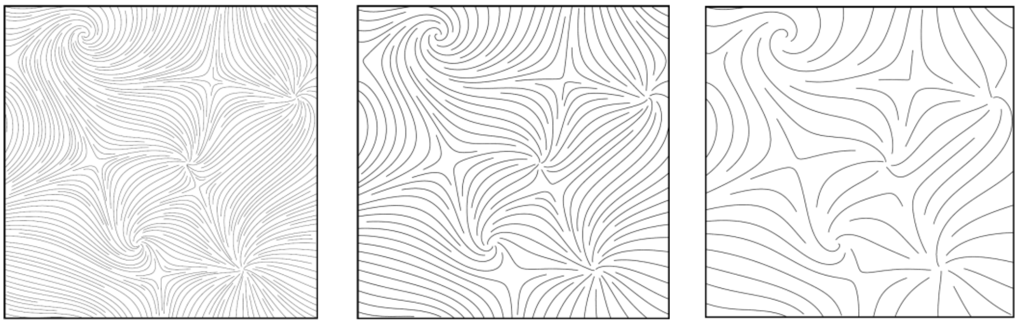
4. RESULTS AND DISCUSSION

We tested our approach for some analytical and computational data sets. The data sets are used to compare random seeding against DCM seeding. The quality of streamlines relays on the coverage, uniformity and continuity. For the continuity, all the streamlines generated by our method are complete streamline, which means the streamlines are the longest of all the streamlines passing through the same seeds. Because there are no standards to compare uniformity and continuity quantitatively, we compare the results with other methods visually.

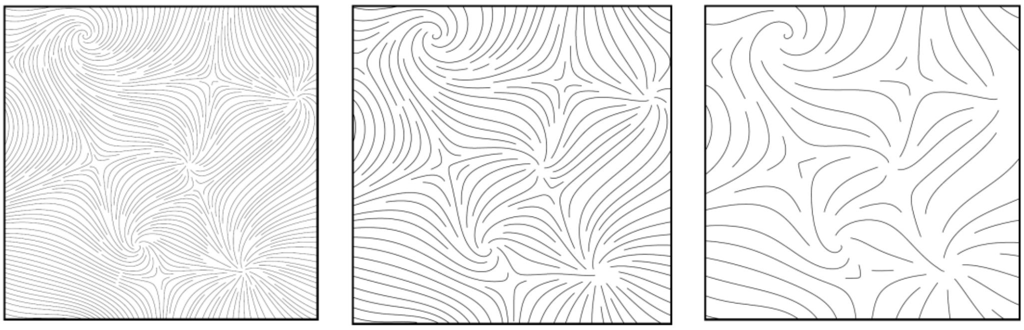
Our results have been generated on a Windows Vista ThinkPad T61p notebook equipped with an Intel

Core2 Duo T7500 2.2GHZ CPU, 3GB Ram, Nvidia Qurdro FX 570M 128M GPU. All the three tests cost no more than 10 seconds including the DCM calculation process which costs most of the time.

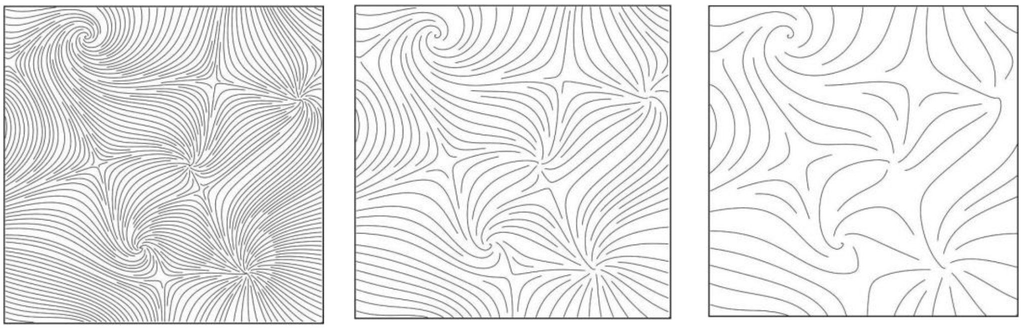
Figure 3 shows the comparison with other methods. The vector field consists of 50*50 vectors. All method have almost the same results with more streamlines. The compared algorithm tends to produce short separated streamline and is much more obvious when using less streamlines. Our method does not require as much uniformity as others do, by which it can capture more features with less streamlines, which is shown in center and right of Figure 3(d).



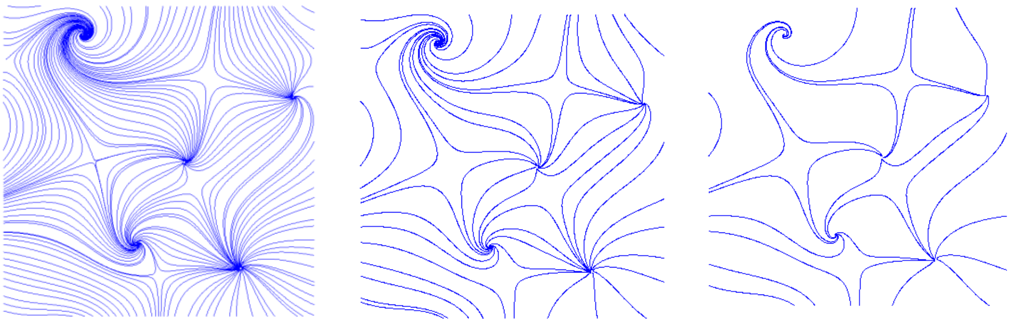
(a) Turk/Banks([Tur96])



(b) Jobard/Lefer([Job97])



(c) Mebarki et al.([Meb05])



(d). Our method

Figure 3 Comparison of streamline placement techniques

Figure 4 shows a slice of a 3D vector field. The vector field consists of 128*128 vectors, which comes from simulation of swirling jet entering fluid. Figure 4(a) and 4(b) show results of our method. The swirl of the vector field is well captured. On the other hand, Figure 4 (c) and (d) show the results of algorithm of Jobard/Lefer. The swirl is not so distinct, for the streamlines are not long enough to reveal the features.

Figure 5 shows comparison with algorithm of Vermal et al. The vector field consists of 70*70 vectors. Figure 5(a) and 5(b) show results of algorithm of Verma. The algorithm does perform well in the critical regions. In other words, the critical regions can not be well represented, especially when fewer streamlines are used. Figure 5(c) and 5(d) show results of our method. Very few streamlines are produced in Figure 5(d), but the critical regions are very clear.

Our algorithm only uses complete streamlines. The long streamlines are preferred in this paper, while discontinuities in the layout with shorter streamlines may impair the impression of a flow field.

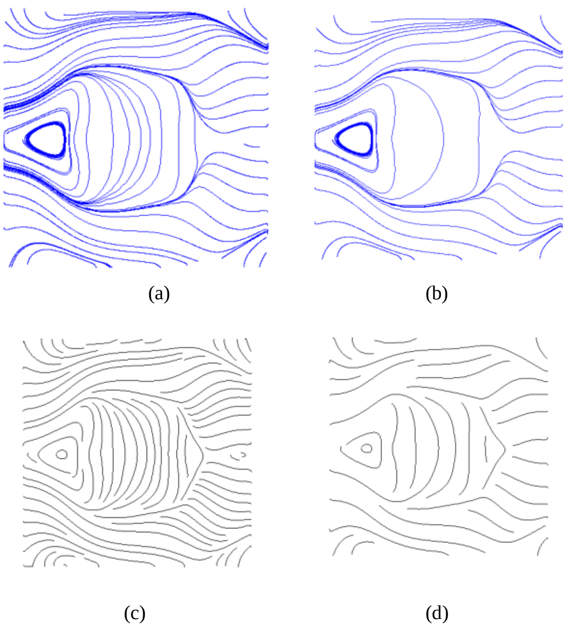


Figure 4 Swirling jet entering fluid at rest.

Our seeding strategy picks position with the maxima of DCM. The greater difference-contribution the position has , the greater the variation is. The position with great variation is picked firstly, such as convergent point. And there are less streamlines in the region with lower difference-contribution, such as in Figure 4(b) while the streamlines in Figure 4(d) are still even almost everywhere.

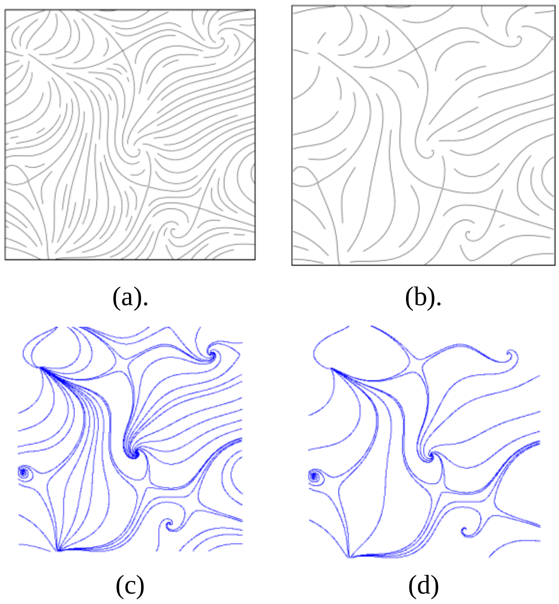


Figure 5. Comparison to feature-based technique

5. CONCLUSION

A DCM seeding strategy is proposed in this paper. We introduced inward and outward contribution of a position as variation measure of the vector field. Then DCM is defined. The streamline starting points are seeded mainly depending on the maxima of the DCM matrix. The generation of each streamline lowers the matrix locally until the given condition is satisfied.

The new approach catches regions with great variation and the vector field can be represented by less streamlines.

6. ACKNOWLEDGEMENTS

We would like to thank Roger Crawfis for providing the tornado dataset, and also University of California Davis for the provision of the swirling jet dataset.

The work described in this paper was supported by Chinese National High-Tech R&D Program Grant (2007AA01Z318, 2007AA01Z159, 2009AA01Z324), National Natural Science Foundation of China (90915010 , 60925007, 60973052, 60703062, 60833007, U0735004), National Basic Research Program of China(2010CB328002).

7. REFERENCES

[Liu06] Liu, Z.: ‘An Advanced Evenly-Spaced Streamline Placement Algorithm’, IEEE Transactions on Visualization and Computer Graphics, 2006, 12, (5), pp. 965-972

[Ver00] Verma, V., D. Kao, and A. Pang. A flow-guided streamline seeding strategy. in IEEE Visualization 2000, pp. 163-170

[Lar04] Laramee, R.S., Hauser, H., Doleisch, H., Vrolijk, B., Post, F.H., and Weiskopf, D.: 'The state of the art in flow visualization: dense and texture-based techniques', Computer Graphics Forum, 2004, 23, (2), pp. 203-221

[Pos03] Post, F.H., Vrolijk, B., Hauser, H., Laramee, R.S., and Doleisch, H.: 'The state of the art in flow visualisation: Feature extraction and tracking', Computer Graphics Forum, 2003, 22, (4), pp. 775-792

[Tur96] Turk, G. and D. Banks. Image-guided streamline placement. in SIGGRAPH 1996. pp. 453-460. New York,USA.

[Mao98] Mao, X.Y., et al. Image-guided streamline placement on curvilinear grid surfaces. in IEEE Visualization '98. 1998. pp. 135-142.

[Job97] Jobard, B. and W. Lefer. Creating evenly-spaced streamlines of arbitrary density. in Visualization in scientific computing '1997, pp. 43-56.

[Job00] Jobard, B., and Lefer, W.: 'Unsteady flow visualization by animating evenly-spaced streamlines', Computer Graphics Forum, 2000, 19, (3), pp. C31-C39.

[Job01] Jobard, B., and Lefer, W.: 'Multiresolution flow visualization', WSCG '2001: Short Communications and Posters, 2001, pp. P34-P37.

[Mat03] Mattausch, O., et al. Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines. in Spring Conference on Computer Graphics. 2003: ACM New York, NY, USA, pp. 213-222.

[Meb05] Mebarki, A., P. Alliez, and O. Devillers. Farthest point seeding for efficient placement of streamlines. in IEEE Visualization 2005, pp. 479-486.

[Ye05] Ye, X.H., Kao, D., and Pang, A.: 'Strategy for seeding 3D streamlines', IEEE Visualization 2005, Proceedings, 2005, pp. 471-478.

[Li07] Li, L.Y., and Shen, H.W.: 'Image-based streamline generation and rendering', IEEE Transactions on Visualization and Computer Graphics, 2007, 13, (3), pp. 630-640.

[Sch07] Schlemmer, M., et al. Priority Streamlines: A context-based Visualization of Flow Fields. in EuroVis07: Joint Eurographics - IEEE VGTC Symposium on Visualization. 2007, pp. 227-234.

[Gre92] Greene, J.M.: 'Locating three-dimensional roots by a bisection method', J. Comput. Phys., 1992, 98, (2), pp. 194-198.

[Hel89] Helman, J., and Hesselink, L.: 'Representation and Display of Vector Field Topology in Fluid-Flow Data Sets', Computer, 1989, 22, (8), pp. 27-36.

[Hel91] Helman, J.L., and Hesselink, L.: 'Visualizing Vector Field Topology in Fluid-Flows', Ieee Computer Graphics and Applications, 1991, 11, (3), pp. 36-46.

[Set99] Sethian, J.A.: 'Fast marching methods', Siam Rev, 1999, 41, (2), pp. 199-235.