

# A Streamline Selection Technique Overlaying with Isosurfaces

Shiho Furuya, Takayuki Itoh

**Abstract** Integration of scalar and vector visualization has been an interesting topic. This paper presents a technique to appropriately select and display multiple streamlines while overlaying with isosurfaces, aiming an integrated scalar and vector field visualization. The technique visualizes a scalar field by multiple semitransparent isosurfaces, and a vector field by multiple streamlines, while the technique adequately selects the streamlines considering reduction of cluttering among the isosurfaces and streamlines. The technique first selects and renders isosurfaces, and then generates large number of streamlines from randomly selected seed points. The technique evaluates each of the streamlines according to their shapes on a 2D display space, distances to critical points of the given vector fields, and occlusion by isosurfaces. It then selects the specified number of highly evaluated streamlines. As a result, we can visualize both scalar and vector fields as a set of view-independently selected isosurfaces and view-dependently selected streamlines.

## 1 Introduction

Scalar and vector visualization techniques have been almost independently evolved in the research field of volume visualization. Meanwhile, it often happens that we want to simultaneously and comparatively visualize both scalar and vector fields in a same physical space. Cluttering is a serious problem while visualizing both scalar and vector fields in a single display space. This paper presents a technique which simultaneously visualizes both scalar and vector fields in a same physical space. We

---

Shiho Furuya  
Ochanomizu University, e-mail: shiho@itolab.is.ocha.ac.jp

Takayuki Itoh  
Ochanomizu University e-mail: itot@is.ocha.ac.jp

(This paper is presented at TopoInVis 2017, 2/27/2027 at Keio University, Japan.)

visualize a scalar field as a set of semitransparent isosurfaces, and a vector field as a set of streamlines. The presented technique features automatic selection of specified number of streamlines from the following three aspects: geometric meaningfulness evaluated by information entropy, distance from critical points, and occlusion by isosurfaces.

This paper is an extended version of our previous poster presentation [2], which additionally introduces subjective evaluations, and a case study with a weather simulation dataset which visualized relationship between an air flow field and high/low pressure regions.

## 2 Related Work

### 2.1 Streamline selection

Automatic streamline selection techniques have been firstly developed for 2D flow visualization, and then extended to 3D visualization.

Mebarki et al. [13] claimed that uniformly placed sets of streamlines can be generated by specifying positions of seed points of streamlines as most distant from previously generated streamlines. Li et al. [11] presented a 2D visualization technique which generates appropriately distant sets of streamlines while controlling the number of streamlines.

Many 3D streamline selection techniques have been recently presented. Ye et al. [21] presented a technique which prepares several templates to specify positions of seed points of streamlines around critical points of 3D flow fields, so that users can appropriately generate streamlines which represent the features of flow fields. Li et al. [10] presented a streamline generation technique which sets seed points on a plane in a 3D space, so that we can avoid cluttering of streamlines. Marchesin et al. [12] extended our idea [2] for view-dependent streamline removal and addition. Several recent studies [9] [19] estimate viewpoints while selecting streamlines in 3D flow fields, so that we can observe the flow fields from the best viewpoint which brings comprehensive visualization results.

However, most of existing streamline selection techniques just attempt to optimize the visualization of flow fields. On the contrary, our technique presented in this paper selects streamlines for simultaneous visualization of scalar and vector fields in a same physical 3D space.

### 2.2 Integrated visualization of scalar and vector fields

We often have 3D flow dynamics simulation datasets which contain both scalar and vector fields in a same physical space. It is important to simultaneously visualize

both scalar and vector fields to understand the flow dynamics. Direct volume rendering is a convenient method for this purpose. It is not usually difficult to control transfer functions of volume rendering to overlay indirect vector field visualization such as arrow plots and streamlines. Such representation has been applied since the early days of visualization research. Crawfis et al. [15] represented scalar and vector fields by using splatting-based volume rendering and particle animations. Hong et al. [4] applied glyphs which represent vector fields while overlaying with volume rendering for scalar fields.

On the other hand, it sometimes happens that indirect volume rendering is more preferable. We may have a situation that visualization results should be stored as indirect datasets (e.g. isosurfaces or iso-contours for scalar fields, streamlines or arrow plots for vector fields). Or, runtime environments which GPU programming is not available are still existing. Combination of indirect visualization techniques for scalar and vector fields is another solution for these situations, as we apply isosurfaces and streamlines in this paper. Actually, there are many studies which applies isosurfaces and streamlines for integrated scalar and vector field visualization [14] [16]. However, these studies did not apply automatic streamline selection techniques.

It is also meaningful to apply isosurfaces to represent flow fields [1]. For example, Laramer et al. [8] applied velocity isosurfaces with streamlines to effectively represent flow fields.

### 3 Presented visualization technique

This section describes the processing flow and technical detail of the presented technique. Following is the brief processing flow of the technique.

- Step 1: Extract critical points of given scalar and vector fields.
- Step 2: Generate semitransparent isosurfaces to represent a scalar field.
- Step 3: Temporality generate large number of streamlines.
- Step 4: Select specified number of streamlines which are highly evaluated based on information entropy, distances to critical points, and occlusion by isosurfaces.

The following subsections describe the technical detail of the each process of the presented technique.

#### 3.1 Critical point extraction

We often intensively visualize around critical points of scalar or vector fields to discover interesting phenomena. In other words, we can often visualize interesting phenomena by generating isosurfaces and streamlines around critical points. For example, in case of visualizing weather simulation results, we can often discover

interesting scalar fields around high (or low) pressure regions, or interesting vector fields around centers of vortices.

Accordingly, we developed a critical point extraction technique for scalar and vector fields, to intensively visualize the scalar and vector values around the critical points. We apply a critical point extraction technique for scalar fields presented by Itoh et al. [5]. This technique extracts local maximum or minimum points of scalar fields by comparing scalar values of grid-points connected by element-edges.

We also apply a critical point extraction technique for vector fields presented by Koyamada et al. [7]. This technique determines that a position  $P(p, q, r)$  is a critical point of a vector field inside a tetrahedral cell, if  $P$  satisfies the following equation:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = M_v^{-1} \begin{pmatrix} -u_3 \\ -v_3 \\ -w_3 \end{pmatrix}, \quad M_v = \begin{vmatrix} u_0 - u_3 & u_1 - u_3 & u_2 - u_3 \\ v_0 - v_3 & v_1 - v_3 & v_2 - v_3 \\ w_0 - w_3 & w_1 - w_3 & w_2 - w_3 \end{vmatrix} \\ 0 \leq p \leq 1, 0 \leq q \leq 1, 0 \leq r \leq 1$$

Here,  $(u_i, v_i, w_i)$  is a vector value at the  $i$ -th grid-point of the current tetrahedral cell. Also, we suppose the current tetrahedral cell can be divided into four sub-tetrahedral regions by connecting  $P$  and each of vertices of the tetrahedral cell. Ratios of volumes of the sub-tetrahedral regions are  $p, q, r$ , and  $(1 - p - q - r)$ .

### 3.2 Visualization of scalar fields

This technique uses the critical points of a scalar field while generating isosurfaces. Our implementation lists scalar values at the extracted critical points and displays as a user interface widget. When a user selects one of the displayed scalar values, the technique generates an isosurface by applying Marching Cubes. Our implementation allows to display semi-transparent isosurfaces while selecting multiple scalar values.

Our current implementation requires manual adjustment the opacity of isosurfaces. As a future work, we would like to automatically adjust by applying automatic transfer function design techniques [6][17].

### 3.3 Visualization of vector fields

#### 3.3.1 Information entropy for streamline selection

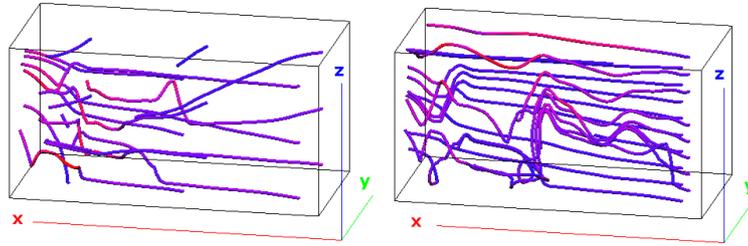
This section describes an automatic streamline selection technique applying information entropy. First, this technique uniformly generates large number of streamlines thoroughly inside an input volume region. Then, the technique calculates information entropy for each of the streamlines applying a technique presented by

Takahashi et al. [18]. We automatically select a specified number of streamlines which have larger information entropy values. Information entropy has been well applied to variety of recent visualization techniques [20]. The following is an equation applied in our technique to calculate information entropy  $E$ :

$$E = -\frac{1}{\log_2(1+m)} \sum_{j=0}^m \frac{d_j}{l_j} \log_2 \frac{d_j}{l_j} \quad (1)$$

Here, we suppose streamlines consist of short segments.  $l_j$  and  $d_j$  are the length of the  $j$ -th segment in the 3D physical and 2D display spaces respectively, and  $m$  is the number of the segments in a streamline.

We can reselect appropriate sets of streamlines by calculating  $E$  for all of the pre-generated streamlines, when the viewpoint is moved or view direction is modified. Our implementation supports drag operations of pointing devices to control the viewpoint and view direction, and execute the automatic streamline selection when the drag operations are finished.



**Fig. 1** Example of streamline generation. (Left) Uniformly generated streamlines. (Right) Streamlines automatically selected based on information entropy.

Figure 1(Left) shows an example of uniformly generated streamlines, and Figure 1(Right) shows an example of streamlines selected based on information entropy, where both examples draw 18 streamlines. Our technique successfully represents the vector field comparing with uniformly generating streamlines.

Furthermore, our streamline selection technique takes into account critical points, occlusion by isosurfaces, and density of streamlines, as described in below sections.

### 3.3.2 Streamline selection with critical points

Often we would like to visualize the vector fields around critical points because interesting phenomena are often found there [7]. However, we experimentally found that  $d_j$  in equation (1) of streamlines around critical points tends to be small and therefore such streamlines are not often selected while simply applying the informa-

tion entropy. Therefore, our implementation optionally selects at least one streamline generated from vicinity of each of critical points.

### 3.3.3 Occlusion by isosurfaces

We would like to avoid un-preferable visualization results that many streamlines are occluded by isosurfaces. Our technique revises the information entropy by the following equation to preferentially select un-occluded streamlines:

$$E' = \left(1 - \frac{m_0}{m}\right)E + \left(\frac{m_0}{m}\right)(1 - \alpha)E, \quad (2)$$

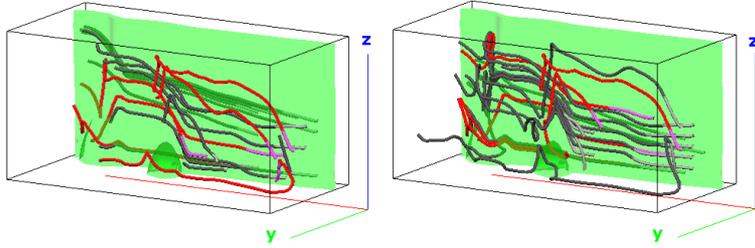
where  $m_0$  is the number of occluded segments of a streamline, and  $\alpha$  is the opacity of an isosurface.

Our implementation displays multiple isosurfaces as described in Section 3.2. Supposing the  $i$ -th segment ( $1 \leq i \leq m$ ) of a streamline is occluded by the  $k$ -th isosurface which is rendered with the opacity value  $\alpha_k$ , we define the weight of the  $i$ -th segment as follows:

$$d_{ik} = (1 - \alpha_k)d_{i(k-1)} \quad (3)$$

It may also happen that streamlines are occluded by multiple isosurfaces. We can calculate the weights of segments of such streamlines by recursively applying equation (3). As a result, we calculate the information entropy by the following equation, supposing  $1 \leq j \leq n_j$  and  $d_{i_0} = d_i$ :

$$E = -\frac{1}{\log_2(1+m)} \sum_{j=0}^m \frac{d_{jn_j}}{l} \log_2 \frac{d_{jn_j}}{l} \quad (4)$$



**Fig. 2** Simultaneous display of isosurfaces and streamlines. (Left) Without calculating weights based on occlusions by isosurfaces. (Right) With calculating weights.

Figure 2 demonstrates the effectiveness of the weights based on occlusions of isosurfaces. Figure 2(Left) shows that many streamlines behind an isosurface is selected, while Figure 2(Right) shows the presented technique successfully selects

more visible streamlines. As a result, our technique could automatically visualize a vortex in the left side of the input volume.

### 3.3.4 Density of streamlines

Our technique based on information entropy has another problem that large number of similar and close streamlines can be selected. Accordingly, we determine if selected streamlines are sufficiently distant each other, so that we can avoid to densely select close streamlines. Our current implementation exclusively selects one of the streamlines which intersect with the same cell, to avoid selecting too close streamlines. However, this implementation does not always work well if the resolution of an input volume is finer comparing with resolution of a display. We would like to implement an image-based streamline density control [10] to solve this problem. Opacity control [3] is another solution to generate more comprehensive flow visualization for such situations.

## 4 Example and evaluation

We implemented the presented technique with GNU C++ and OpenGL, and conducted subjective evaluation with the visualization results. Participants of the evaluation were 12 university students majoring computer graphics and visualization. We showed them pairs of visualization results and asked to select one of them. Tables in the below sections show the number of participants who selected the corresponding visualization results.

### 4.1 Effectiveness of streamline selection

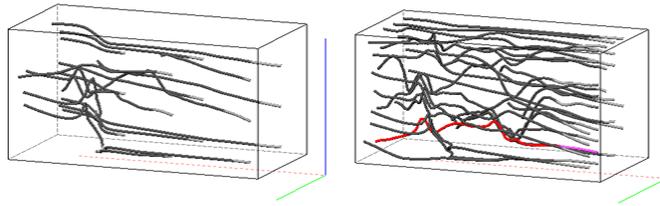
We generated visualization results with three viewpoints. We used two implementations, where one uniformly generates streamlines, and the other displays streamlines automatically selected by the presented technique. Then, we asked participants to subjectively evaluate the results. Figure 3 shows the results. Streamlines generated around critical points are drawn in red, while others are drawn in black.

We asked participants to select the visualization result which satisfies:

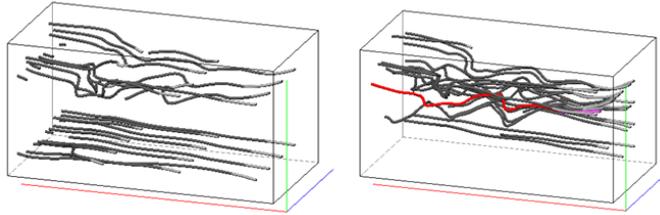
- Red streamlines are more visible.
- Black streamlines are less visible.

Table 1 shows the statistics of the evaluations of participants.

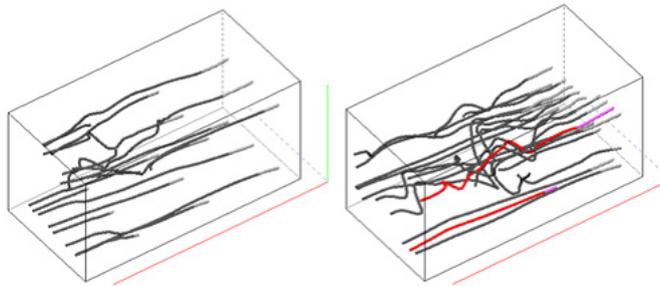
The presented technique received preferable evaluations, especially with viewpoints A and C. Following are typical comments for these results by the presented technique:



(Upper-left) Uniformly generated streamlines from viewpoint A. (Upper-right) Streamlines selected by our technique from viewpoint A.



(Center-left) Uniformly generated streamlines from viewpoint B. (Center-right) Streamlines selected by our technique from viewpoint B.



(Lower-left) Uniformly generated streamlines from viewpoint C. (Lower-right) Streamlines selected by our technique from viewpoint C.

**Fig. 3** Effectiveness of streamline selection.

- It is easier to understand the vector field since the presented technique selects longer streamlines.
- The presented technique well represents local phenomena such as undulation of winds.
- The presented technique shows larger number of red streamlines.

On the other hands, four participants did not prefer the result by the presented technique with viewpoint B. Their comments are as follows:

- Visualization by the presented technique is too dense.
- The result with uniformly selecting streamlines is simpler.

**Table 1** Subjective evaluation for effectiveness of streamline selection.

	Uniformly generated	Automatically selected
Viewpoint A	1	11
Viewpoint B	4	8
Viewpoint C	2	10

## 4.2 Effectiveness of critical point extraction

We also asked participants to evaluate the results shown in Figure 4, as our previous questions introduced in Section 4.1. This figure compares the visualization results whether critical points are extracted or not.

**Table 2** Subjective evaluation for effectiveness of critical point extraction.

	Without critical points	With critical points
Viewpoint A	4	8
Viewpoint B	4	8
Viewpoint C	4	8

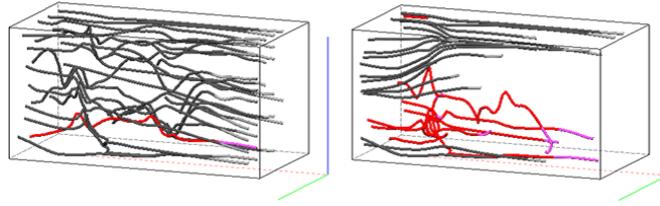
Table 2 shows the statistics of the evaluations of participants. Extracting critical points and generating at least one streamline around each of the critical points, we could show more characteristic streamlines, and therefore received more preferable evaluations. On the other hand, several participants mentioned that too many streamlines drawn in red might make difficult to understand the vector fields.

## 4.3 Effectiveness of detection of occlusion by isosurfaces

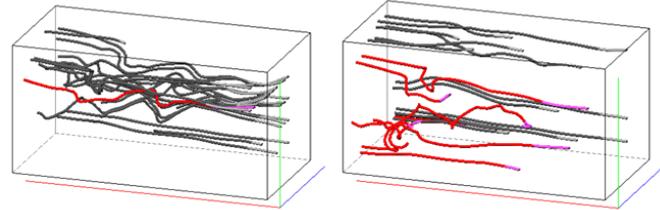
Finally, We asked participants to evaluate the results shown in Figure 5, as our previous questions introduced in Sections 4.1 and 4.2. This figure compares the visualization results whether occlusion by isosurface is detected or not.

**Table 3** Subjective evaluation for effectiveness of occlusion detection.

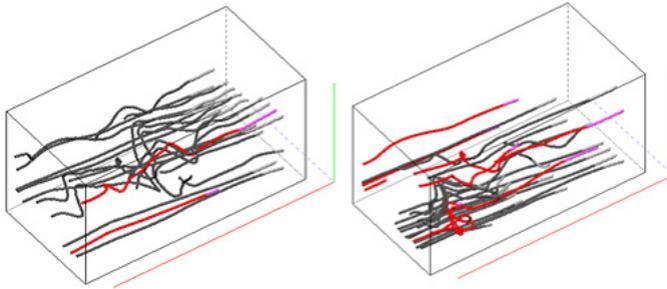
	Without occlusion detection	With occlusion detection
Viewpoint A	8	4
Viewpoint B	6	6
Viewpoint C	5	7



(Upper-left) Without critical point extraction, from viewpoint A. (Upper-right) With critical point extraction, from viewpoint A.



(Center-left) Without critical point extraction, from viewpoint B. (Center-right) With critical point extraction, from viewpoint B.



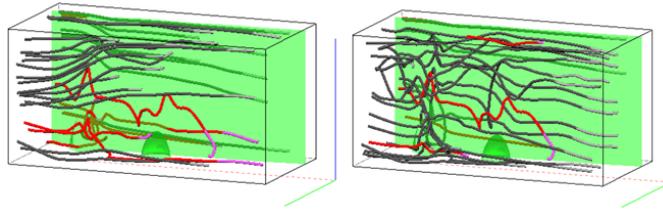
(Lower-left) Without critical point extraction, from viewpoint B. (Lower-right) With critical point extraction, from viewpoint B.

**Fig. 4** Effectiveness of critical point extraction.

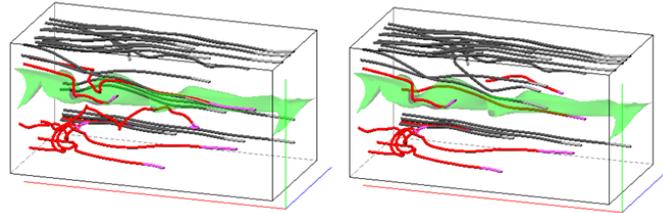
Table 3 shows the statistics of the evaluations of participants. Our technique received relatively preferable evaluation while applying viewpoint C. Following are major comments for the results with viewpoint C detecting occlusion by isosurfaces:

- Red streamlines are not occluded by an isosurface.
- Relationship between an isosurface and streamline is more comprehensive.

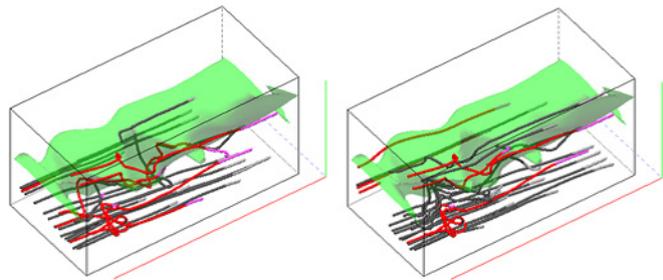
On the other hand, our technique received relatively poor results with viewpoint A. Following are major comments for the results with viewpoint A detecting occlusion by isosurfaces:



(Upper-left) Without detecting occlusion by isosurfaces, from viewpoint A. (Upper-right) With detecting occlusion by isosurfaces, from viewpoint A.



(Center-left) Without detecting occlusion by isosurfaces, from viewpoint B. (Center-right) With detecting occlusion by isosurfaces, from viewpoint B.



(Lower-left) Without detecting occlusion by isosurfaces, from viewpoint C. (Lower-right) With detecting occlusion by isosurfaces, from viewpoint C.

**Fig. 5** Effectiveness of occlusion detection.

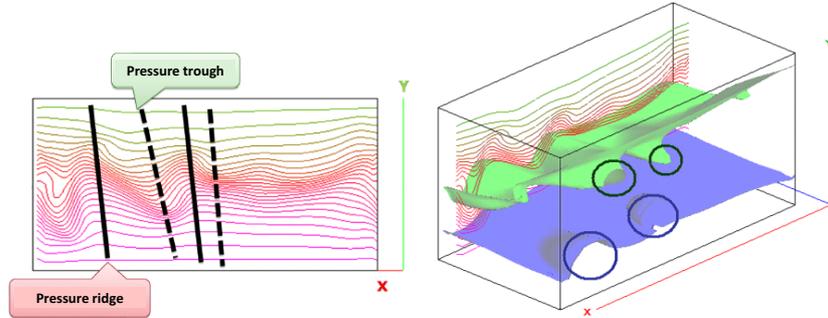
- Visualization result got too complicated because too many streamlines are displayed in front of an isosurface.
- Red streamlines are occluded by black streamlines rather than an isosurface.

## 5 Visualization of weather simulation data

This section introduces a visualization of weather simulation dataset formed as a regular grid. Number of cells are  $80 \times 40 \times 80$ , where each grid-point has scalar values

(air pressure and temperature) and a vector value (air flow). Longitude and latitude are assigned to the X- and Y-axes, while height is assigned to the Z-axis.

First, we extracted critical points from a scalar (pressure) field. It contained 13 local maximum, 9 local minimum, and 25 saddle points. This dataset simulates an air flow just before an extratropical cyclone is formed.



**Fig. 6** (Left) Air pressure on a XY-plane. (Right) High and low pressure represented by isosurfaces.

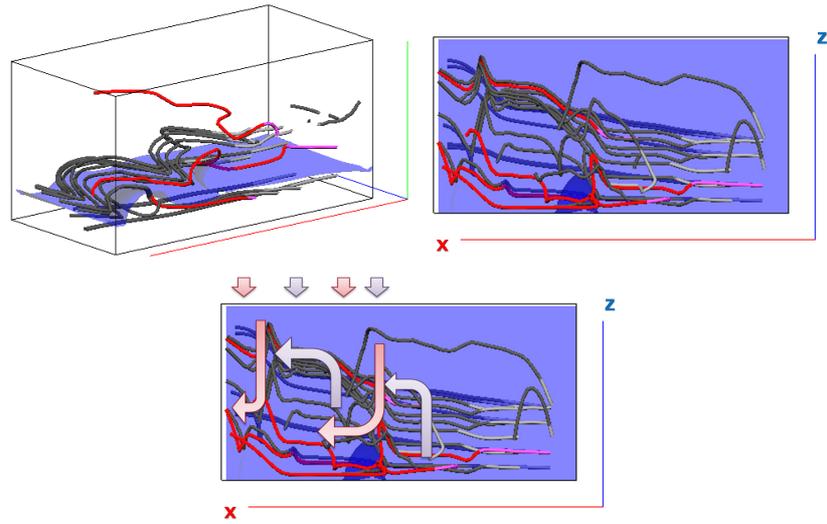
Figure 6(Left) shows iso-contour lines on a XY-plane. Pressure ridge and trough are drawn as solid and dotted lines respectively. In this result, pressure ridge is developed when westerlies swerves toward north, while pressure trough appears when westerlies swerves toward south.

Figure 6(Right) shows a blue isosurface whose isovalue is slightly smaller than the pressure value at one of local maximum points, and a green isosurface whose isovalue is slightly larger than the pressure value at one of local minimum points. High pressure regions appears where the blue isosurface curves, indicated by circles drawn in dark blue in Figure 6(Right). Low pressure regions appears where the green isosurface curves, indicated by circles drawn in dark green in Figure 6(Right).

Figure 7(Upper) shows a visualization of air flow applying streamlines around a high pressure region. This visualization displays streamlines around critical points in red, and the others in black. Start points of the streamlines are drawn in subtle colors so that we can recognize these directions. Left and right examples in Figure 7(Upper) display different sets of streamlines because our implementation recalculates information entropy when the viewpoint or view direction is switched.

Figure 7(Lower) illustrates pressure ridge and trough shown in Figure 6(Left) in addition to an isosurface and streamlines visualized in Figure 7(Upper). Blue arrows depict pressure ridge, while pressure trough are illustrated by red arrows. Downdraft is observed around pressure ridge, and updraft is observed around pressure trough.

High pressure regions are often developed when air temperature in the sky is much lower than one on the ground. We can observe the relationship between high pressure regions and air temperature by switching the scalar field from pressure to temperature and then visualizing the temperature applying isosurfaces.



**Fig. 7** (Upper) Visualization of air flow around a high pressure region. (Lower) Visualization of air flow around pressure ridge and trough.

## 6 Conclusion

This paper presented an integrated visualization of scalar and vector fields applying an automatic streamline selection technique. We select preferable streamlines from the following three aspects: geometric meaningfulness applying information entropy, distance from critical points, and occlusion by isosurfaces. We conducted subjective evaluation with university student participants and concluded that information entropy and distance from critical points were especially effective factors of our technique. We also introduced the effectiveness of our technique with a case study applying a weather simulation dataset.

The following are our potential future issues:

### **Improvement for volume dataset containing large number of critical points.**

The dataset we used as a case study just contained approximately 50 critical points in its scalar and vector fields. It is difficult to deal with all the critical points if there are hundreds or even thousands in a dataset. For example, it is not realistic to generate isosurfaces whose isovalues are close to values of all of thousands of local maximum or minimum points. Or, it is also meaningless to generate streamlines around all of thousands of vortices. We may need to select reasonable number of important critical points [17] as a preprocess in such cases.

**Extension for time-varying datasets.** Our technique would select streamlines independently for each timestep, if we applied the current implementation to time-varying volume datasets. This may cause flickering and prevent the understanding

of time-varying air flow. We would like to solve this problem by seamlessly selecting streamlines for each timestep of time-varying volume datasets.

## Acknowledgement

We appreciate Prof. Tetsuya Kawamura and Ms. Fumi Yasuda to provide us a weather simulation dataset.

## References

1. M. Edmunds, R. S. Laramee, G. Chen, N. Max, E. Zhang, C. Ware, Surface-Based Flow Visualization, *Computers & Graphics*, 36(8), 974-990, 2012.
2. S. Furuya, T. Itoh, A Streamline Selection Technique for Integrated Scalar and Vector Visualization, *IEEE Visualization, Poster Session*, 2008.
3. T. Gunther, C. Ross, H. Theisel, Opacity Optimization for 3D Line Fields, *ACM Transactions on Graphics (SIGGRAPH 2013)*, 32(4), 2013.
4. L. Hong, X. Mao, A. Kaufman, Interactive Visualization of Mixed Scalar and Vector Fields, *IEEE Visualization*, 240-247, 1995.
5. T. Itoh, K. Koyamada, Automatic Isosurface Propagation by Using an Extrema Graph and Sorted Boundary Cell Lists, *IEEE Transactions on Visualization and Computer Graphics*, 1(4), 319-327, 1995.
6. G. Kindlmann, J. W. Durkin, Semi-automatic Generation of Transfer Functions for Direct Volume Rendering, *IEEE symposium on Volume visualization*, 79-86, 1998.
7. K. Koyamada, T. Itoh, Seed Specification for Displaying a Streamline in an Irregular Volume, *Engineering with Computer*, 14, 73-80, 1998.
8. R. S. Laramee, D. Weiskopf, J. Schneider, H. Hauser, Investigating Swirl and Tumble Flow with a Comparison of Visualization Techniques, *IEEE Visualization*, 51-58, 2004.
9. T.-Y. Lee, O. Mishchenko, H.-W. Shen, R. Crawfis, View Point Evaluation and Streamline Filtering for Flow Visualization, *IEEE Pacific Visualization Symposium*, 83-90, 2011.
10. L. Li, H.-W. Shen, Image Based Streamline Generation and Rendering, *IEEE Transactions on Visualization and Computer Graphics*, 630-640, 2007.
11. L. Li, H.-H. Hsieh, H.-W. Shen, Illustrative Streamline Placement and Visualization, *IEEE Pacific Visualization Symposium*, 79-86, 2008.
12. S. Marchesin, C.-K. Chen, C. Ho, K.-L. Ma, View-Dependent Streamlines for 3D Vector Fields, *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 1578-1586, 2010.
13. A. Mebarki, P. Alliez, O. Devillers, Farthest Point Seeding for Efficient Placement of Streamlines, *IEEE Visualization*, 479-486, 2005.
14. D. Merhof, M. Meister, E. Bingol, C. Nimsky, G. Greiner, Isosurface-Based Generation of Hulls Encompassing Neuronal Pathways, *Stereotactic Functional Neurosurgery*, 87(1), 2009.
15. R. A. Crawfis, N. Max, Texture Splats for 3D Scalar and Vector Field Visualization, *IEEE Visualization*, 261-266, 1993.
16. P. Schaefer, M. Gampert, N. Peters, Streamline Segment Topology in the Vicinity of Stagnation Points in Turbulent Flows, *15th International Symposium on Flow Visualization*, 2012.
17. S. Takahashi, Y. Takeshima, I. Fujishiro, Topological Volume Skeletonization and Its Application to Transfer Function Design, *Graphical Models*, Vol. 66, No. 1, 24-49, 2004.
18. S. Takahashi, I. Fujishiro, Y. Takeshima, T. Nishita, A Feature-Driven Approach to Locating Optimal Viewpoints for Volume Visualization, *IEEE Visualization*, 495-502, 2005.

19. J. Tao, J. Ma, C. Wang, C.-K. Shene, A Unified Approach to Streamline Selection and View-point Selection for 3D Flow Visualization, *IEEE Transactions on Visualization and Computer Graphics*, 19(3), 393-406, 2013.
20. L. Xu, T.-Y. Lee, H.-W. Shen, An Information Theoretic Framework for Flow Visualization, *IEEE Transactions on Visualization and Computer Graphics*, 16(6), 1216-1224, 2010.
21. X. Ye, D. Kao, A. Pang, Strategy for Seeding 3D Streamlines, *IEEE Visualization*, 471-478, 2005.