Journal of Oceanology and Limnology
Vol. 37 No. 6, P. 2025-2036, 2019
https://doi.org/10.1007/s00343-019-8263-1

Dynamic visual simulation of marine vector field based on LIC—a case study of surface wave field in typhoon condition*

LIU Zhendong[1, 3], LIU Haixing[1, 2, 3, **], SU Tianyun[1, 2, 3, **], JIA Zhen[1, 3], LI Xinfang[1, 3], ZHOU Lin[1, 3], SONG Zhuanling[1, 3]

[1] *First Institute of Oceanography* (*FIO*), *Ministry of Natural Resources* (*MNR*), *Qingdao 266061*, *China*

[2] *Laboratory for Regional Oceanography and Numerical Modelling*, *Qingdao National Laboratory for Marine Science and Technology*, *Qingdao 266237*, *China*

[3] *National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology*, *Qingdao 266061*, *China*

Received Sep. 25, 2018; accepted in principle Dec. 30, 2018; accepted for publication Feb. 20, 2019

Abstract Line integral convolution (LIC) is a useful visualization technique for a vector field. However, the output image produced by LIC has many problems in a marine vector field. We focus on the visual quality improvement when LIC is applied in the ocean steady and unsteady flow field in the following aspects. When a white noise is used as the input in a steady flow field, interpolation is used to turn the discrete white noise into continuous white noise to solve the problem of discontinuity. The "cross" high-pass filtering is used to enhance the textures of streamlines to be more concentrated and continuity strengthened for each streamline. When a sparse noise is used as the input in a steady flow field, we change the directions of background sparse noise according to the directions of vector field to make the streamlines clearer and brighter. In addition, we provide a random initial phase for every streamline to avoid the pulsation effect during animation. The velocities of vector field are encoded in the speed of the same length streamlines so that the running speed of streamlines can express flow rate. Meanwhile, to solve the problem of obvious boundaries when stitching image, we change the streamline tracking constraints. When a white noise is used as an input in an unsteady flow field, double value scattering is used to enhance the contrast of streamlines; moreover, the "cross" high-pass filtering is also adopt instead of two-dimensional high-pass filtering. Finally, we apply the above methods to a case of the surface wave field in typhoon condition. Our experimental results show that applying the methods can generate high-quality wave images and animations. Therefore, it is helpful to understand and study waves in typhoon condition to avoid the potential harm of the waves to people's lives and property.

Keyword: line integral convolution (LIC); wave data visualization; steady and unsteady marine flow field

1 INTRODUCTION

With the continuous development of marine exploration methods, the ability of humans to acquire marine environmental data has been continuously improved. As a result, the amount of accumulated data is constant increasing. These marine flow field data that change at any time have become a significant theoretical basis for revealing marine phenomena, preventing natural disasters, and studying marine dynamics mechanisms. Therefore, it is urgent to find a tool that can effectively help researchers analyze and understand the laws contained in these data sets. Vector field visualization technology can successfully visualize the massive and complicated ocean data by means of a graphic image. When the technology is applied to the marine, it can express these marine data intuitively and accurately, which can lay a good foundation for analyzing marine laws and forecasting marine disasters.

Images of vector fields should faithfully represent

* Supported by the National Key Research and Development Program of China (No. 2016YFC1402000)
** Corresponding authors: liuhx@fio.org.cn; sutiany@fio.org.cn

the characteristics, e.g. sources and sinks, vortices, critical points, and separation lines. However, the standard visualization methods, such as arrows, icons, field lines, field ribbons, is constrained by spatial resolution, therefore none of them can describe the global flow features. By contrast, interactive texture-based vector field visualization, due to advantages of inherent compactness and high-quality texture, can reveal almost all the details of the flow field. Hence, it is more suitable for expressing the marine vector field with dramatic variations in direction.

Texture-based vector field visualization mainly includes two methods: spot noise (van Wijk, 1991) and LIC (Cabral and Leedom, 1993). Spot noise generates a texture by distributing a set of intensity functions, or spots, over the domain, but this method is not suitable for the visualization of vector fields with relatively severe changes. Line integral convolution (LIC), uses a low pass filter to perform one-dimensional convolution on the noise image in order to produce a texture image highly correlated with the flow field. LIC is more suited for the visualization of critical points, which is a key element in conveying the flow topology. Moreover, compared to spot noise, LIC can display all the details of the vector field more clearly. Therefore, LIC is very suitable to be applied in the marine flow field to observe and study the variations of ocean elements.

This paper is organized as follows. In Section 2, we briefly introduce the LIC and previous work for 2D steady and unsteady flow field. In Section 3, we apply LIC in the marine vector field and propose specific improvements for the problems in the application. In Section 4, we show the results obtained by using our methods to a case of surface wave field during the typhoon. Finally, we summarize and conclude in Section 5.

2 METHOD

2.1 Line integral convolution

LIC is an effective texture synthesis technique to visualize steady and unsteady flow fields. The effectiveness of the output images produced by LIC comes from two types of coherence: spatial and temporal coherence.

In the steady flow field, the LIC maintains the spatial coherence by the method of value gathering along the streamlines to obtain the value of the pixel. Meanwhile, the temporal coherence is established by shifting the filter phase to allow the texture to move along the streamlines. Additionally, both the white noise and sparse noise can be used as background noise. When white noise is used as background noise, the orientation of the vector field must be expressed by animation, while the static picture generated when sparse noise is used as background noise can express the orientation of the vector field.

In the unsteady flow field, the LIC can achieve the spatial and temporal coherence by using the methods of value scattering and successive feed-forward. The created animation can present the variations of a vector field over time.

2.2 Previous work

The researches of LIC-based 2D flow field visualization have developed LIC to several directions: (1) enhancing texture contrast between streamlines, (2) adding direction clues, (3) expressing velocity magnitude, (4) allowing real-time display, and (5) forming smooth and clear animation. Next, we will introduce the researches in steady and unsteady 2D vector field visualization based on LIC.

2.2.1 Steady flow field

Fast LIC (Stalling and Hege, 1995) is a method to speed up the LIC computation. It minimizes the total number of streamlines to be computed and exploits similar convolution integrals along a single streamline and thus reuses parts of the convolution computation from neighboring streamline texels. The multi-frequency LIC (Kiu and Banks, 1996) uses multi-frequency noise instead of white noise as the input noise. As a result, long and fat streaks indicate regions of the flow with higher velocity magnitude. Enhanced LIC (Okada and Kao, 1997) presents a double LIC method to enhance the texture contrast of the output image; it also uses colored texture images to highlight flow separation and reattachment boundaries. The enhanced fast LIC (Hege and Stalling, 1998) experiments with higher order filter kernels in order to enhance the quality of the resulting LIC textures. It also can enhance a user's perception of the magnitudes and direction of the flow. Multivariate LIC (Urness et al., 2003) presents an extension to fast LIC that incorporates a new coloring scheme that can be used to incorporate multiple 2D scalar and vector attributes. OLIC (Wegenkittl et al., 1997) and FROLIC (Wegenkittl and Gröller, 1997) adopt a sparse noise texture and a ramp-like convolution kernel to let the output image indicate the orientations. Animated FROLIC (Berger and Gröller, 2000) achieves an

animation of the result via a color-table and is based on the observation that only the colors of the FROLIC disks need to be changed. Iterative twofold convolution (Weiskopf, 2009) is proposed as an efficient high-quality two-stage filtering method. The first stage employs a compact filter, based on Lagrangian particle tracing; the second stage applies iterative alpha blending to implement a large-scale exponential filter. OGR LIC (Matvienko and Krüger, 2015) considers the spectral properties of the dense flow visualization process as an integral operator defined in a local curvilinear system aligned with the flow. It achieves adaptive local spatial frequency control in flow visualization images not relying on noise injection.

2.2.2 Unsteady flow field

Curvilinear grids and unsteady LIC (Forssell and Cohen, 1995) adopt pathline tracing instead of streamline tracing, as a result, it can successfully modify LIC to visualize unsteady flow field. UFLIC (Shen and Kao, 1997, 1998) consists of a time-accurate value depositing scheme and a successive feed-forward method, which can maintain the spatial and temporal coherence, so it can effectively trace the flow's global features over time. UFLIC requires considerable time to generate each frame due to the huge amount of pathline integration that is computed for particle value scattering, so AUFLIC (Liu and Moorhead II, 2002; Liu and Moorhead II, 2005) is presented. It reuses pathlines in the value scattering process to reduce computationally expensive pathline integration in order to speed up display. In addition, many accurate parallel implementations of UFLIC (Li et al., 2006; Ding et al., 2015) are presented to visualize real-time unsteady flows with high spatial and temporal coherence.

To improve evaluating visualization quality, the theory of human visual perception is applied to the evaluation of visualization results and the improvement of visualization effects (Ma and Guo, 2018). Besides, LIC is also widely used in other fields, such as medical science, pencil hatching generation and so on (Höller et al., 2016; Li et al., 2016; Höller et al., 2017; Kong et al. 2018; Zheng et al., 2018).

3 IMPROVEMENT

In this section, by analyzing the problems when applying the LIC method to the marine vector field, we will introduce our modification methods to better apply LIC to the steady and unsteady marine flow field.

3.1 Steady flow field

3.1.1 White noise as input noise

3.1.1.1 "Cross" high-pass filtering

When the white noise is convoluted by the LIC to obtain the output image, for the sake of enhancing the contrast of the output image texture, the traditional method is to perform two-dimensional high-pass filtering on the output image. Two-dimensional high-pass filtering has a sharpening effect on the image. The two-dimensional high-pass filter matrix evolved from the Laplacian operator. The evolution can be derived as:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}.$$

However, since the two-dimensional high-pass filtering is isotropic high-pass filtering, it does not take into account the directionality of the vector field, so the resulting image texture is not well contrasted and the continuity is not strong. Therefore, we design the "cross" high-pass filtering to improve this problem. The principle of "cross" high-pass filtering is to highlight the boundaries of streamlines and smooth the output texture. "Cross" high-pass filtering can be decomposed into 1D vertical filtering and 1D parallel filtering, which are perpendicular and parallel to the speed direction of any point in the vector field respectively. 1D vertical filtering can enhance the textures of streamlines more concentrated and directional and emphasize the boundaries of streamlines since it can reduce the effect of streamline boundary texture on the center streamline. 1D parallel filtering can be used to strengthen continuity for the textures of each streamline because it can narrow the gap between adjacent textures on the streamline.

The calculation formula of the image after high-pass filtering is as followed:

$$T_{out}=T_{pre}\times M,$$

where T_{out} is the texture value of the pixel of the final output image, T_{pre} is the texture value of the pixel of the preliminary image, M is the filter matrix.

Figure 1 shows the specific process of "cross" high-pass filtering. Firstly, the method of 1D parallel filtering is as followed. V_p is the instantaneous speed of point A on the streamline, and D and E are the two sampling points in the direction, L_p is the sampling distance of this direction. The filter kernel of this direction is a matrix of [1/3, 1/3, 1/3]. Similarly, the

method of 1D vertical filtering is as followed. V_v is perpendicular of the speed direction of point A, and B, C are the sampling points of this direction, and L_v is the sampling distance in this direction. The filter kernel is a matrix of [-1, 2, -1]. The matrix of "cross" filtering can be obtained by combing the two filter kernel matrices. The "cross" high-pass filter matrix of point A is as followed:

$$M=\begin{bmatrix} 0 & M_B & 0 \\ M_D & M_A & M_E \\ 0 & M_C & 0 \end{bmatrix}=\begin{bmatrix} 0 & 1 & 0 \\ 1/3 & 7/3 & 1/3 \\ 0 & 1 & 0 \end{bmatrix},$$

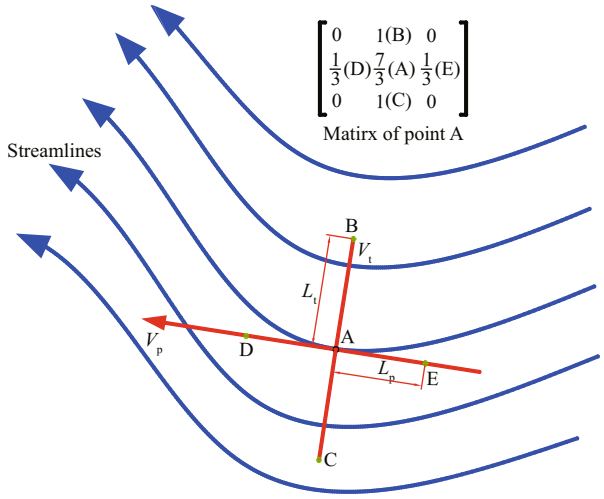

Fig.1 The process of "cross" high-pass filtering, the upper right corner is the filter matrix of point A

where M_A, M_B, M_C, M_D, M_E are the weight values of the five points A, B, C, D and E in Fig.1, respectively. Moreover, the filter kernel matrix M will rotate as the speed direction of the point. For example, if the V_p of any point is perpendicular to the horizontal line, the matrix of this point is formed by rotating the matrix M of point A 90 degrees.

1D parallel filtering is used to homogenize texture along the streamline, thus the sampling distance L_p is set to 1 so that the sampling pixel points are adjacent to the current pixel point on the same streamline. 1D vertical filtering can avoid the texture interaction between different streamlines to make the streamline tighter and centralized. The effect produced by different L_v is shown in the Fig.2. It can be seen from the image that the streamline gap in the texture image increases as L_v increases, and when L_v is too large, the streamline gradually becomes blurred. Therefore, it is significant to choose appropriate L_v, which can display the features in the image more clearly and intuitively. In this paper, we choose the value of L_v to be 2.

Figure 3 shows the original image, the image with two-dimensional high-pass filtering and the image with "cross" high-pass filtering (L_v=2). As shown in Fig.3, compared with the streamlines processed by the two-dimensional high-pass filtering, the boundaries of the streamlines processed by the "cross" high-pass filtering are more obvious, and the streamlines are smoother and more continuous.

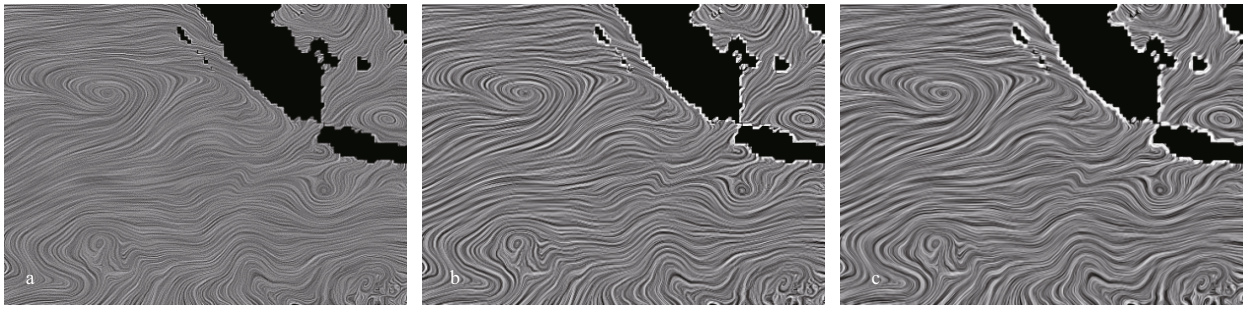

Fig.2 The image with "cross" high-pass filtering, the "L_v" is 1, 3, 5 (from left to right)

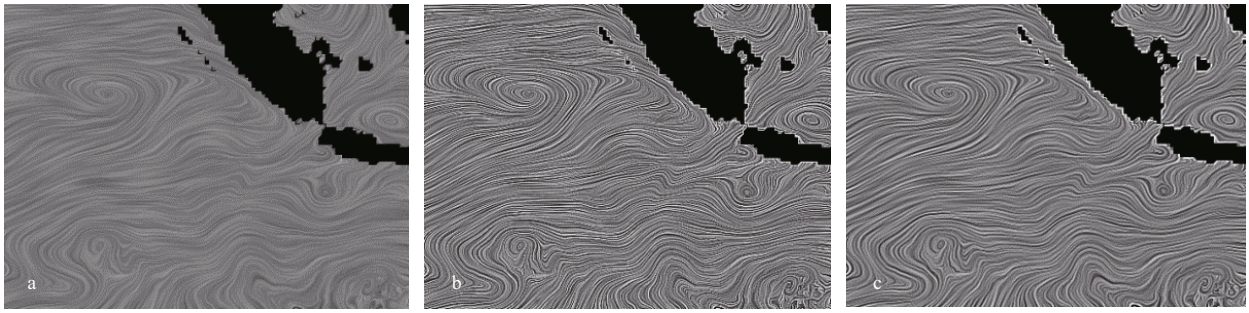

Fig.3 The original image (a), the image with two-dimensional high-pass filtering (b), and the image with "cross" high-pass filtering (c)

3.1.1.2 Continuous background white noise

The streamline for each pixel starts at the center of this pixel and moves out in the positive and negative directions. Each pixel that passed through by the streamline is given a certain weight by the convolution kernel to convolve the white noise. In this way, we can obtain the final texture value of the pixel of the output image. The streamlines by tracking the adjacent pixels have similar trajectories. However, since the white noise is a discrete random noise point when the background white noise points have been passed through by the streamlines of similar trajectories, are greatly different, the image is prone to the discontinuity.

As shown in Fig.4, the two streamlines are formed by tracking the adjacent pixel points A and B. The white noise points have been passed through by the two streamlines which have similar trajectories, are very different. Therefore, the output texture values of the two points obtained by convolution are also different, resulting in poor continuity of the image. To solve this problem, we use the method of interpolating background white noise to change the discrete white noise into the continuous white noise; therefore, it can narrow the distinction in texture values of different white noise points that have been passed through by streamlines that have similar trajectories. We use the method of bilinear interpolation using the formulas as follows:

$$f(x,y_1)=\frac{x_2-x}{x_2-x_1}f(Q_{11})+\frac{x-x_1}{x_2-x_1}f(Q_{21}),$$

$$f(x,y_2)=\frac{x_2-x}{x_2-x_1}f(Q_{12})+\frac{x-x_1}{x_2-x_1}f(Q_{22}),$$

$$f(x,y)=\frac{y_2-y}{y_2-y_1}f(x,y_1)+\frac{y-y_1}{y_2-y_1}f(x,y_2),$$

where (x, y) is the point to be inserted, $f(Q_{11})$, $f(Q_{21})$, $f(Q_{12})$, $f(Q_{22})$, are the texture values of the four adjacent pixels of the lower left, lower right, upper left, and upper right of the (x, y) point, respectively. (x_1, y_1), (x_2, y_1), (x_1, y_2), (x_2, y_2) are the coordinates of the center point of the four pixels.

Before using our method, the texture inside the red rectangle has obvious anti-aliasing, as shown in Fig.5a. After adopting our method, the anti-aliasing phenomenon in the red rectangular frame is obviously improved due to the decrease of the background noise texture values of the streamlines tracked by adjacent seed points, as shown in Fig.5b.

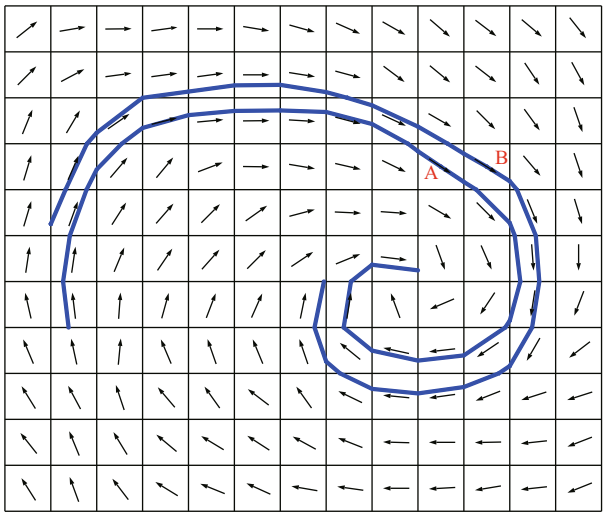

Fig.4 The streamlines formed by tracking adjacent pixel a and b, the tracking step size is 13

3.1.2 Sparse noise as input noise

3.1.2.1 Changing the direction of sparse noise

When the background noise is sparse noise, the initial directions of the noise groups in the background noise are horizontal or vertical, while the directions of the streamlines in the flow field may exist in all directions. Therefore, the directions of the noise groups are irrelevant to the directions of the flow field, which may result in a wide streamline and blurred color. We use a method of changing the directions of the noise groups of the sparse noise according to the directions of the vector field so that the noise groups can be roughly arranged according to the flow directions of the vector field (Fig.6). The directions of the local vector field are calculated by formula as follows:

$$v_{\text{avg}}=\frac{ff[0]+\cdots+ff[8]}{9},$$

where v_{avg} is the average vector size of the sparse noise group, $ff[0]..ff[8]$ are the vector values of the central portion of the sparse noise group, respectively.

As shown in Fig.7, when the noise groups of the sparse noise are uniformly arranged horizontally, the tilted streamlines have significant gaps and blurred textures. After applying our method, since the directions of the noise groups have changed according to the directions of the vector field, the formed streamlines can become more slender and brighter. As a result, the effect of visualization turns better.

Pseudo code is as follows:

Generate sparse background noise

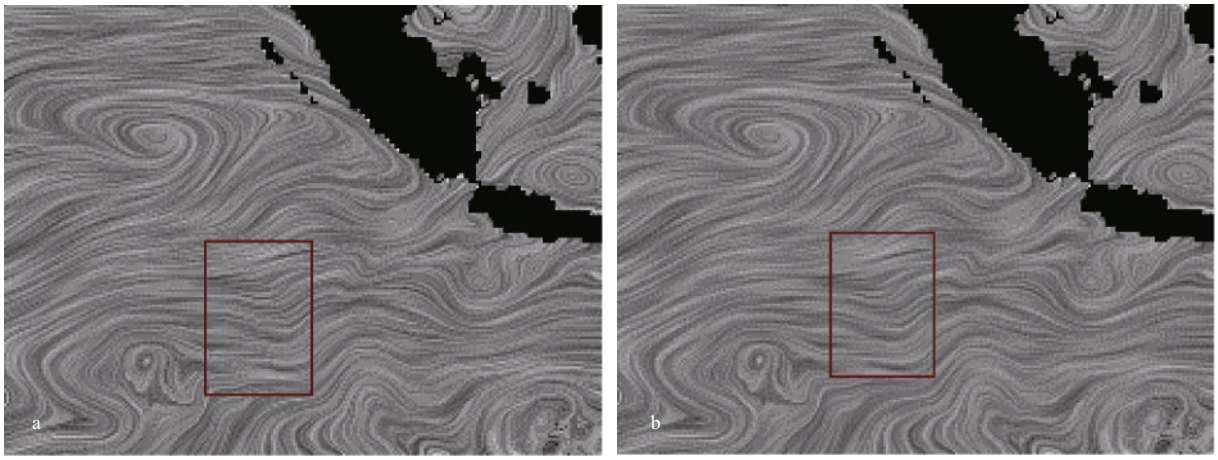

Fig.5 The image (a) is formed by discrete white noise, the image (b) is formed by continuous white noise

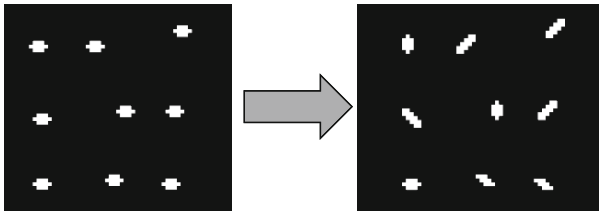

Fig.6 The left image shows the original sparse noise group, the right image shows the sparse noise group improved by our method

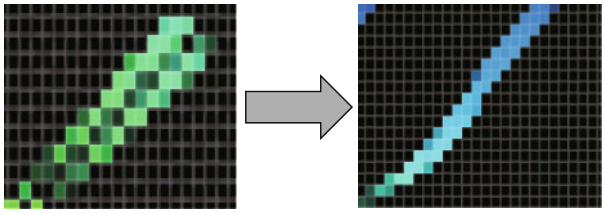

Fig.7 The left and right graphs are streamline diagrams generated by applying sparse noise before and after improvement

{

Setting a series of sparse noise groups for each arrangement direction;

For (areas where each sparse background noise group can move)

{

Calculating the average direction v_{avg} of the sparse noise group vector direction;

Select a sparse noise group corresponding to the direction according to v_{avg};

}

}

3.1.2.2 Setting different initial phases

When the background noise is sparse noise, the loop animation can be achieved by simply phase shifting the ramp-like convolution kernel for each frame accordingly. However, at the loop junction, the loop animation has a significant abrupt displacement. This is because after the ramp-like convolution kernel is extended to a periodic function, the convolution kernel at the junction is not continuous. Its weight value is abruptly changed from the maximum value to zero. The same phenomenon occurs at the same time, causing a sudden shift in the loop animation. The key to the problem of loop animation is that all streamlines have the phenomenon of abrupt displacement at the same time, so we solve this problem by applying a random initial phase for each streamline in the image, causing the phenomenon of abrupt displacement for each streamline at a different time. Therefore, we can form a coherent loop animation.

3.1.2.3 The speed of the streamline movement expresses the flow rate

The velocity of the flow field is encoded in the length of the pixel traces (Wegenkittl et al., 1997), so the flow velocity can be expressed by the length of the streamline. Nevertheless, the animation did not work well especially for the areas with low flow rates. In this paper, we relate the magnitude of the flow velocity to the speed of the streamline movement. As a result, we express the flow velocity by the running speed of each streamline of the same length in the animation.

The method can be divided mainly into two stages, in which we use different conditions for restricting streamline tracking. In the first stage, we used a method to limit the tracking time of streamlines. Since each pixel has a different vector velocity value, each streamline will have a different motion distance during the same motion time, so that we can form streamlines of different lengths according to the

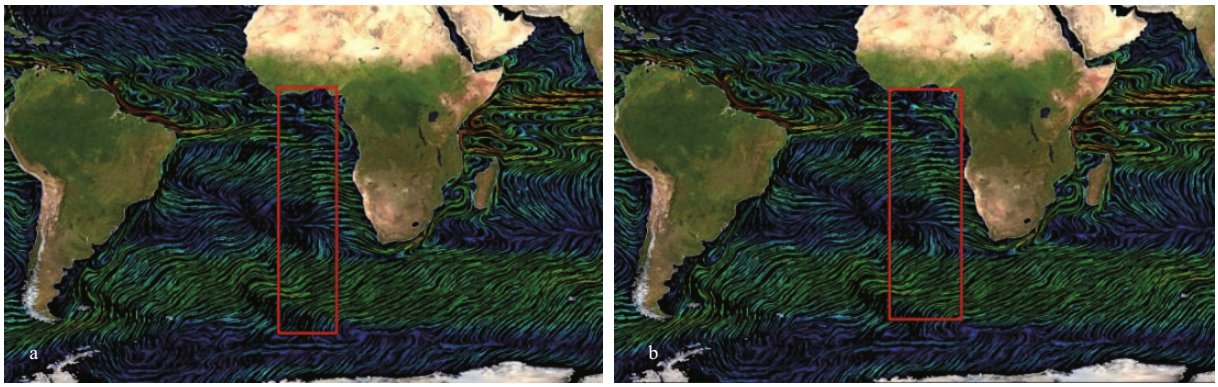

Fig.8 Current image before stitching (a), stitched current image (b)

The boundary inside the red box in (a) is improved in (b)

different motion distance during the same time. In the second stage, we use a method of limiting the streamline tracking step size, so that we can form streamlines of the same length. According to the differences in the length of each streamline formed in the first stage, the phase change rate of the convolution kernel is given for the corresponding same length streamline formed at the second stage. The phase change rate of the convolution kernel can change the movement speed of the streamlines. Hence, the velocity of the vector field is closely combined with the running speed of the streamline to achieve the purpose of expressing the flow rate by using the running speed of the streamline of the same length during animation.

Pseudo code is as follows:

```
For (areas where each sparse background noise group can move)
{
  Calculating the moving distance of each streamline at the same time;
}
Generating background noise;
Generating a ramp-like convolution kernel function;
For (each pixel of the image)
{
  The corresponding convolution kernel change rate is assigned according to the length of the streamline;
  Tracking streamline in the positive and negative vector direction some fixed distance;
  Convolution integral calculation to obtain the output texture value of the pixel;
}
```

3.1.2.4 Stitching image

When turning a planar image of the global current into a spherical image, there is a clear boundary when the two sides of the planar image are joined together. This phenomenon is caused by the stop of streamline tracking when the streamline is hit by the left and right side boundaries of the image. To solve this problem, we change the constraints of the streamline tracking, as a result, the streamlines can be tracked back and forth at the left and right side boundaries of the image, so that the sides of the image can be completely stitched together (Fig.8). Figure 9 is the final spherical current image without boundary. The principle is as follows:

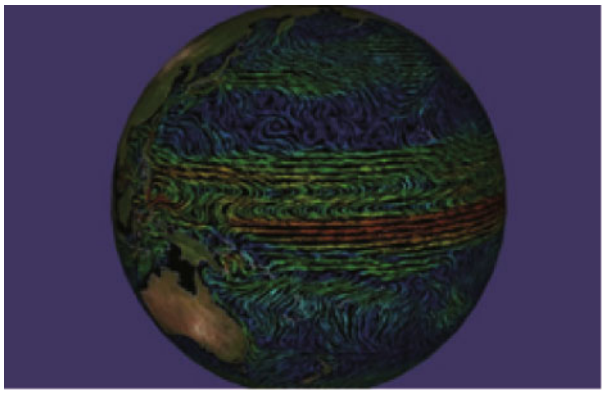

Fig.9 The spherical current image without boundary is formed by our method

$$P_{\text{lon,lat}} \in \{P_{0,0}, P_{0,1}, \cdots, P_{m,n}\}$$

if Direction=1

$$P_{\text{lon=0,lat}} \rightarrow P_{\text{lon=360,lat}}$$

if Direction=0

$$P_{\text{lon=360,lat}} \rightarrow P_{\text{lon=0,lat}},$$

where subscripts lon and lat are longitude and latitude, respectively. The direction is the tracking direction, 0 is the positive tracking, 1 is the negative tracking, and m, n is the resolution of the image in latitude and longitude.

As shown in Fig.8, the red frame area is the

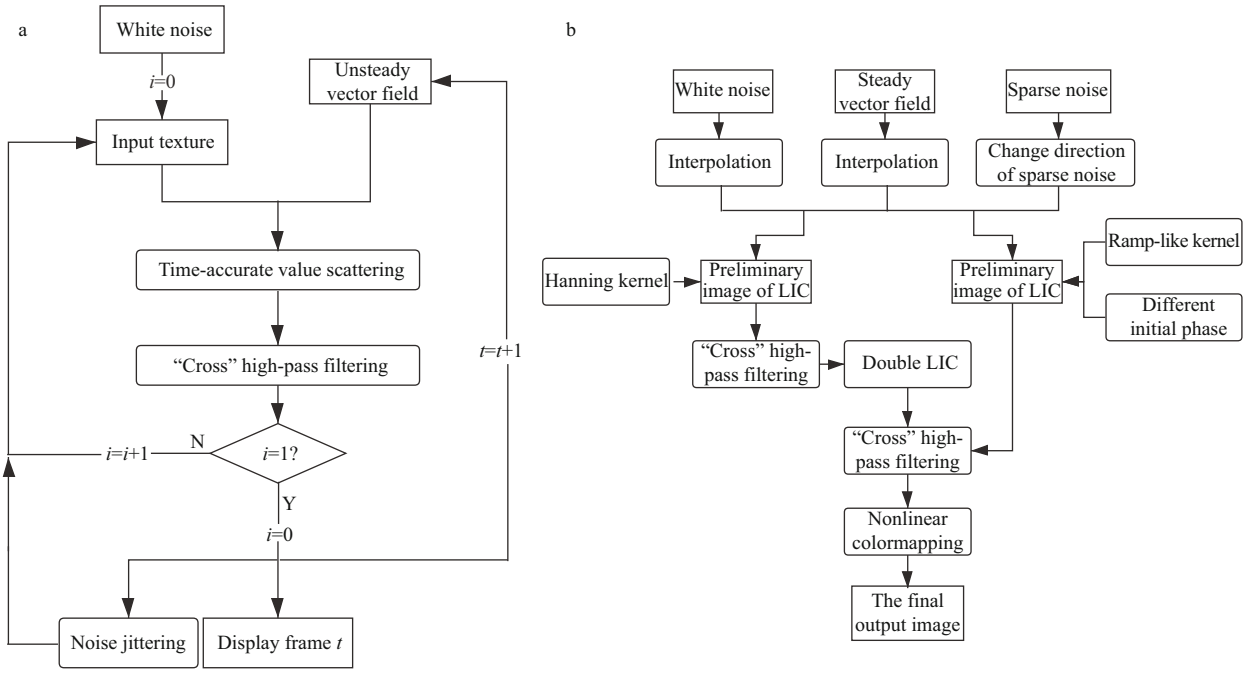

Fig.10 The flow charts of our algorithm for unsteady (a) and steady (b) flow field

boundary junction. There is a clear boundary in the red frame and the streamlines on the left and right sides are disconnected in Fig.8a. The boundary in the red box disappears and the streamlines are continuous without interruption after adopting our proposed method as shown in Fig.8b.

3.2 Unsteady flow field

UFLIC (Shen and Kao, 1998) is a convolution algorithm that uses a time-accurate value scattering scheme to model the texture advection. In addition, the output image from the previous convolution is used as the input texture for the next time step in order to maintain the temporal coherence. Before the output image is used as input at the next step, noise jitter high pass filter should be applied to the output image to enhance the contrast among flow lines.

In order to further enhance the contrast of the streamlines in the image, we use a method of double value scattering for each time step, which is similar to the double LIC in the steady flow field. Our method works as follows (Fig.10a): after the output image of any time step is processed by the high-pass filter, it is still used as the input for this time step to perform a value scattering again, and then the obtained output image processed by high-pass filtering is the final output image of this time step. We jitter the final output image with the original white noise texture to avoid aliasing. Afterwards, it is used as the input for the next time step. Similarly, we can get the final output image of each time step in turn. In addition, we use the "cross" high-pass filtering mentioned above instead of two-dimensional high-pass filtering, which can enhance the textures more concentrated and strength the continuity of each streamline. The image produced by UFLIC and our method is shown in Fig.11.

3.3 Time and space complexity

3.3.1 Time complexity

In the steady flow field, when white noise or sparse noise is used as the background noise, we use the Hanning convolution kernel and the ramp-like convolution kernel, respectively. The classical algorithm of LIC is to traverse all the pixels, in turn, so the time complexity is linear with n, where n represents the total number of points in the vector field (the meaning of n is the same as the following). As a result, the time complexity of the classical algorithm of LIC is O(n). Because our improved methods do not add nested loops based on the classical algorithm, the time complexity of our methods is still O(n).

In the unsteady flow field, because value scattering, calculating pixel values of output texture and high-pass filtering are linear with n, the time complexity of our method is also O(n).

3.3.2 Space complexity

In the steady and unsteady flow field, we apply for

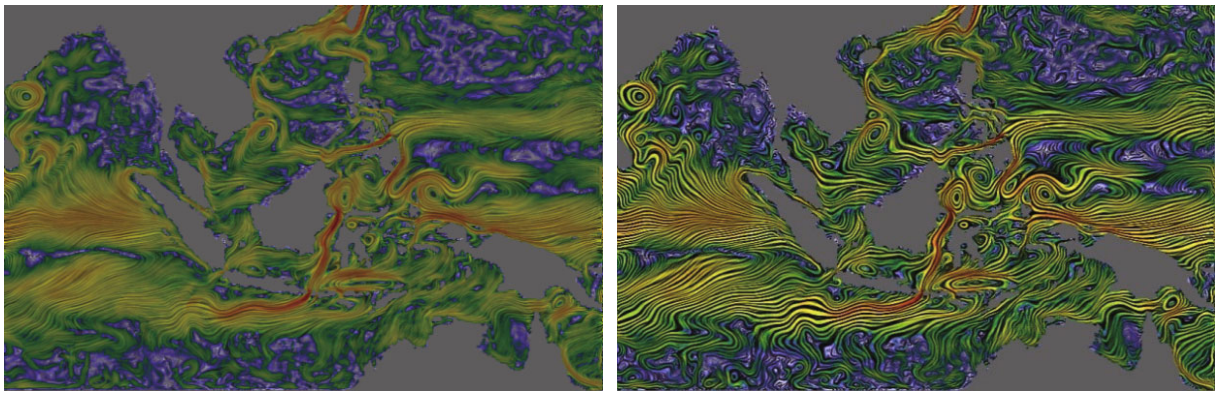

Fig.11 The image generated by UFLIC (left) and our method (right)

arrays based on the total number of points in the vector field, so their space complexity is O(n).

4 IMPLEMENTATION AND RESULT

Every year, a large number of economic losses and casualties are caused by the waves formed by the typhoon in our country. Related researches have shown that 90% of the natural destructive power of the sea comes from the waves, and only 10% of the destructive power comes from the wind. The wind shelter that is often said at sea is actually avoiding the waves. Because any shelter cannot avoid the wind, it can only avoid the waves. According to statistics, the shipwrecks caused by the huge waves at sea account for 60% to 80% of the world's shipwrecks. More than 1 million ships in the world have been sunk in the waves. Given that, preventing the disasters caused by the waves is of vital importance. This paper aims at using our methods to provide support during marine disasters in an effort to prevent and decrease the losses for country and people in the future.

In this paper, we use the wave data sets of the Bohai Bay during the typhoon Damrey that occurred in 2012. Based on the above-improved methods, we use white noise or sparse noise as input in the steady vector field, and the white noise as input in the unsteady flow field to visualize the ocean waves. The flow chart is shown in Fig.10.

The computer environment used for visual simulation in this paper is Intel core i7-2760QM 2.4G P4 CPU, 8GB DDR3 memory and Nvidia NVS 4200 graphics card. The operating system is Windows 7 64-bit SP1 and the development environment is Visual Studio 2010.

We conducted research based on wave data sets generated by Typhoon Damrey. The tropical storm Damrey was launched on the evening of July 28, 2012. At 20 o'clock on the 28th, its center was located on the northwest Pacific Ocean surface of 1 330 km south-southeast of Tokyo, Japan. At 8 o'clock on August 1, it was strengthened into a typhoon. It entered the East China Sea in the early morning of August 2, and then landed onto the coast of Jiangsu Province at 21:30. Soon afterwards, it started moving to the Shandong Province. At 9:00 on August 3, it weakened into a tropical storm in Laiwu City, Shandong Province. At 2 o'clock on August 4, it entered the Bohai Sea, then it weakened into a tropical depression in the northwestern part of Bohai Sea at 8 o'clock, and its residual circulation landed in Liaoning.

We use the wave data sets around the Bohai Sea during the typhoon Damrey. The specific time of the data sets are from midnight on August 1 to 23 on August 3, and the latitude is 37°–41.1°N, the longitude is 117.6°–124°E. We simulate the wave images by using our methods when the white noise or sparse noise is the background noise in the steady flow field (Fig.12), and when the white noise is the background noise in the unsteady flow field (Fig.13). We use textures to represent the direction of the waves and colors to indicate the height of the waves. The images in Fig.13 are sequentially generated over time during a typhoon. As can be seen from the Figs.12 & 13, our method can clearly show the direction and height of waves at any position in the vector field compared to the traditional method of using arrows to indicate the direction of the waves. Since it does not miss any detail of the waves in the vector field, it can help the ships at any position in the sea to know the direction and height of the waves in real time. At the same time, it can also help ships determine the specific location and movement of the disastrous waves, so that the ships can formulate a reasonable navigation route to avoid a shipwreck accident. In addition, our methods can also display the direction change of the coastal

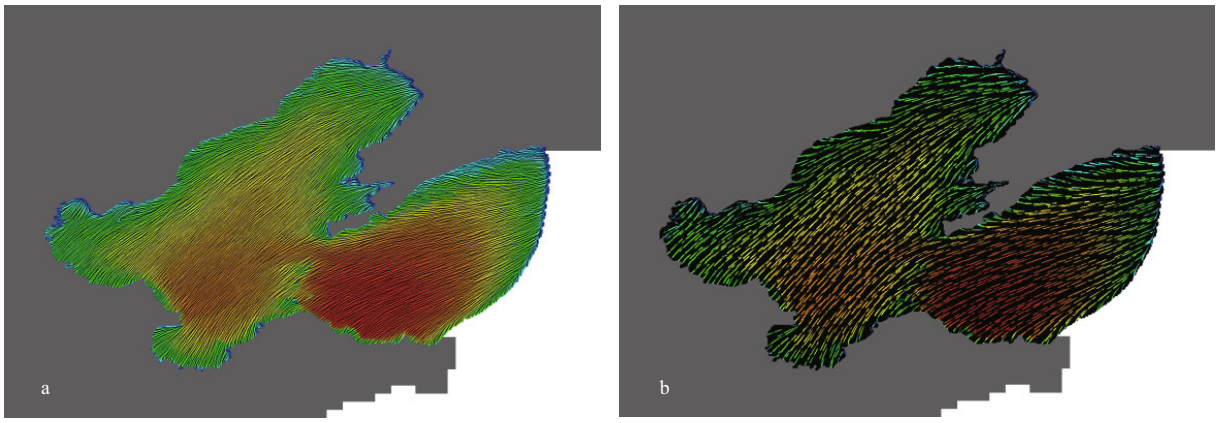

Fig.12 The images of surface waves are generated in a steady flow field based on white noise (a) or sparse noise (b)

The height of waves is expressed by the color.

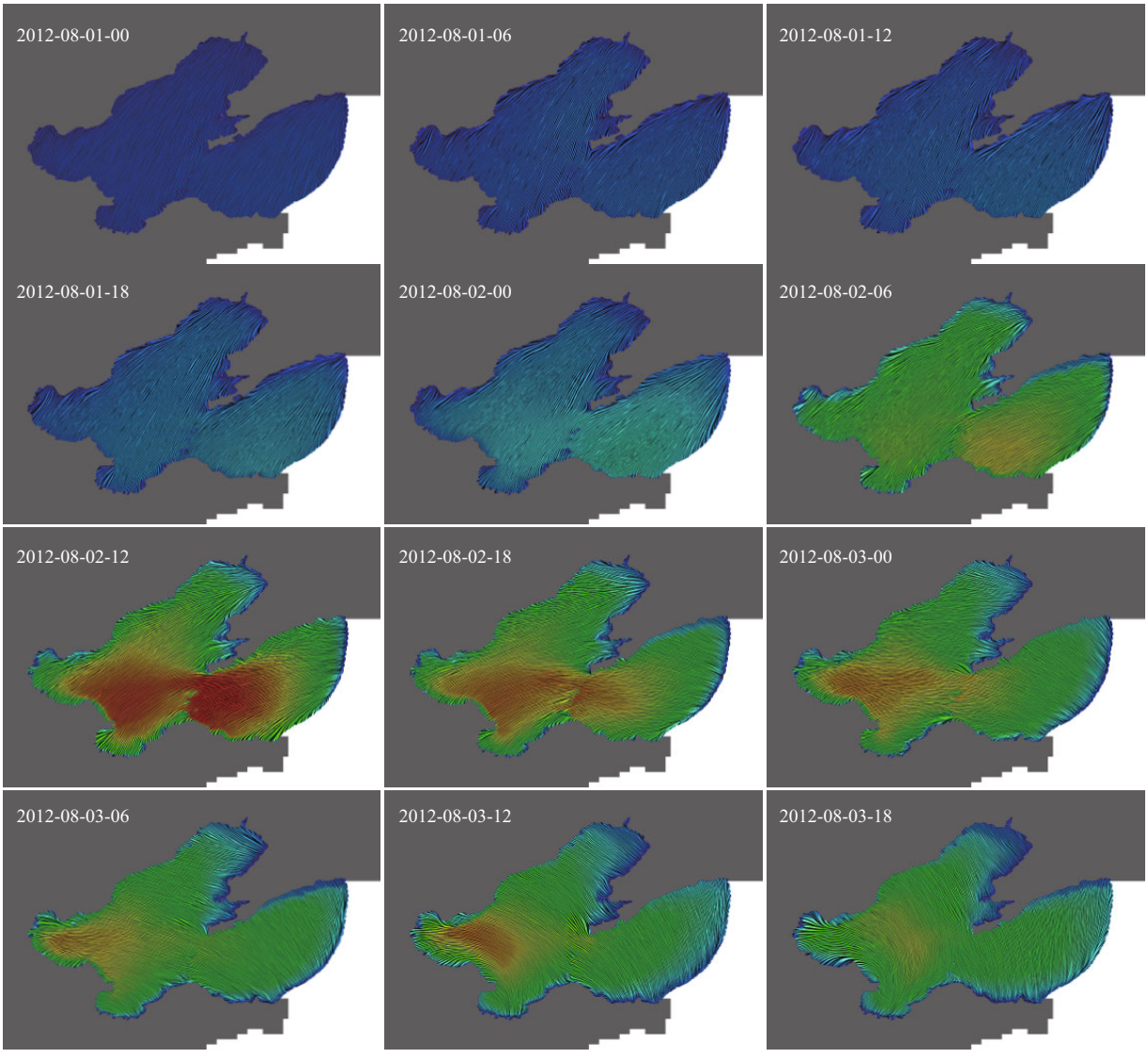

Fig.13 The images of surface waves, the height of waves is expressed by the colors

The images are formed in turn by the methods in the unsteady flow field over time.

waves in the whole coastal area all the time, thus it can provide better support for the wave prevention work in the coastal areas.

The images generated by our visualization methods can clearly show every detail of the wave vector field. Consequently, it is possible to provide assistance in reducing the damage caused by the waves in the typhoon to people's lives and property.

5 CONCLUSION

High-quality vector field visualization can help users comprehend the hidden rules behind a large number of messy data more clearly and intuitively, especially for marine vector field which is containing large-scale data. Based on the LIC, this paper studies the improvements in the visual quality of the ocean flow field. When the white noise is the background noise in the steady flow field, we use the methods of interpolating background white noise and "cross" high-pass filtering in order to enhance the textures of streamlines more concentrated and strengthen continuity for each streamline, as a result, the output image can better express the streamlines and characteristics of vector fields. When the sparse noise is used as input in the steady flow field, improvements are made by changing the directions of background sparse noise and setting different initial phases. As a consequence, the streamlines of the output image are slender and the loop animation is clear and smooth. We also encoded the velocities in the speed of streamline movement in order to express the velocities of flow field by the running speed of same length streamlines. Furthermore, we change the streamline tracking constraints to solve the problem of obvious boundaries when stitching image. When the white noise is the background noise in the unsteady flow field, we use the methods of double value scattering for each time step and "cross" high-pass filtering so that we can obtain clearer texture image. Based on the wave data sets of Bohai during typhoon Damrey, the output images generated by our methods can display all the details of a vector field, and the animation can clearly and intuitively show the changes of the height and orientation of waves over time. It is useful to avoid disaster damage to people's lives and property. It is believed that our methods are of practical help to marine disaster precaution and mitigation.

6 DATA AVAILABILITY STATEMENT

The datasets analyzed during the current study are available from the corresponding author on reasonable request.

7 ACKNOWLEDGMENT

We thank Professor LIU Zhanping of the Old Dominion University for providing technical assistance.

References

Berger S, Gröller E. 2000. Color-table animation of fast oriented line integral convolution for vector field visualization. *In*: Proceedings of the 8th International Conference in Central Europe on Computers Graphics. Research Division of Computer Graphics Research Publications, Bohemia, Plzen. p.4-11.

Cabral B, Leedom L C. 1993. Imaging vector fields using line integral convolution. *In*: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques. ACM, Anaheim, CA. p.263-270, https://doi.org/10.1145/166117.166151.

Ding Z A, Liu Z P, Yu Y, Chen W. 2015. Parallel unsteady flow line integral convolution for high-performance dense visualization. *In*: Proceedings of 2015 IEEE Pacific Visualization Symposium. IEEE, Hangzhou, China. p.25-30, https://doi.org/10.1109/PACIFICVIS.2015.7156352.

Forssell L K, Cohen S D. 1995. Using line integral convolution for flow visualization: curvilinear grids, variable-speed animation, and unsteady flows. *IEEE Transactions on Visualization and Computer Graphics*, **1**(2): 133-141, https://doi.org/10.1109/2945.468406.

Hege H C, Stalling D. 1998. Fast LIC with piecewise polynomial filter kernels. *In*: Hege H C, Polthier K eds. Mathematical Visualization. Springer, Heidelberg, Berlin. p.295-314.

Höller M, Ehricke H H, Synofzik M, Klose U, Groeschel S. 2017. Clinical application of fiber visualization with LIC maps using multidirectional anisotropic glyph samples (A-Glyph LIC). *Clinical Neuroradiology*, **27**(3): 263-273, https://doi.org/10.1007/s00062-015-0486-8.

Höller M, Klose U, Gröschel S, Otto K M, Ehricke H H. 2016. Visualization of MRI diffusion data by a multi-kernel LIC approach with anisotropic glyph samples. *In*: Linsen L, Hamann B, Hege H C eds. Visualization in Medicine and Life Sciences III. Springer, Cham, Switzerland. p.157-177.

Kiu M H, Banks D C. 1996. Multi-frequency noise for LIC. *In*: Proceedings of the Seventh Annual IEEE Visualization 1996. IEEE, San Francisco, CA, USA. p.121-126, https://doi.org/10.1109/VISUAL.1996.567784.

Kong Q Y, Sheng Y, Zhang G X. 2018. Hybrid noise for LIC-based pencil hatching simulation. *In*: Proceedings of 2018 IEEE International Conference on Multimedia and Expo (ICME). IEEE, San Diego, CA, USA. p.1-6, https://doi.org/10.1109/ICME.2018.8486527.

Li G S, Tricoche X, Hansen C. 2006. GPUFLIC: interactive

and accurate dense visualization of unsteady flows. *In*: Eurographics/IEEE-VGTC Symposium on Visualization. IEEE, Lisboa, Portugal. p.29-34, https://doi.org/10.2312/VisSym/EuroVis06/029-034.

Li P K, Zang Y, Wang C, Li J, Cheng M, Luo L, Yu Y. 2016. Road network extraction via deep learning and line integral convolution. *In*: Proceedings of 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, Beijing, China, https://doi.org/10.1109/IGARSS.2016.7729408.

Liu Z P, Moorhead II R J. 2002. AUFLIC: an accelerated algorithm for unsteady flow line integral convolution. *In*: Proceedings of IEEE TCVG Symposium on Visualization. Eurographics Association, Barcelona, Spain. p.43-52, http://dx.doi.org/10.2312/VisSym/VisSym02/043-052.

Liu Z P, Moorhead II R J. 2005. Accelerated unsteady flow line integral convolution. *IEEE Transactions on Visualization and Computer Graphics*, **11**(2): 113-125, https://doi.org/10.1109/TVCG.2005.21.

Ma Y Y, Guo Y F. 2018. Visualization of vector field using line integral convolution based on visual perception. *In*: Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control. ACM, Stockholm, Sweden, https://doi.org/10.1145/3284557.3284709.

Matvienko V, Krüger J. 2015. Explicit frequency control for high-quality texture-based flow visualization. *In*: Proceedings of 2015 IEEE Scientific Visualization Conference (SciVis). IEEE, Chicago, IL, USA. p.41-48, https://doi.org/10.1109/SciVis.2015.7429490.

Okada A, Kao D L. 1997. Enhanced line integral convolution with flow feature detection. *In*: Proceedings of SPIE 3017, Visual Data Exploration and Analysis IV. SPIE, San Jose, CA, United States. p.206-217, https://doi.org/10.1117/12.270314**.**

Shen H W, Kao D L. 1997. UFLIC: A line integral convolution algorithm for visualizing unsteady flows. *In*: Proceedings of Visualization '97 (Cat. No. 97CB36155). IEEE, Phoenix, AZ, USA. p.317-323.

Shen H W, Kao D L. 1998. A new line integral convolution algorithm for visualizing time-varying flow fields. *IEEE Transactions on Visualization and Computer Graphics*, **4**(2): 98-108, https://doi.org/10.1109/2945.694952.

Stalling D, Hege H C. 1995. Fast and resolution independent line integral convolution. *In*: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. ACM, Los Angeles, CA, USA. p.249-256, https://doi.org/10.1145/218380.218448.

Urness T, Interrante V, Marusic I, Longmire E, Ganapathisubramani B. 2003. Effectively visualizing multi-valued flow data using color and texture. *In*: Proceedings of the 14th IEEE Visualization 2003. IEEE, Seattle, WA, USA. p.115-122, https://doi.org/10.1109/VISUAL.2003.1250362.

van Wijk J J. 1991. Spot noise texture synthesis for data visualization. *ACM SIGGRAPH Computer Graphics*, **25**(4): 309-318, https://doi.org/10.1145/127719.122751.

Wegenkittl R, Gröller E, Purgathofer W. 1997. Animating flow fields: rendering of oriented line integral convolution. *In*: Proceedings of Computer Animation 1997. IEEE, Geneva, Switzerland. p.15-21, https://doi.org/10.1109/CA.1997.601035.

Wegenkittl R, Gröller E. 1997. Fast oriented line integral convolution for vector field visualization via the Internet. *In*: Proceedings of IEEE Visualization 1997. IEEE, Phoenix, AZ, USA. p.309-316, https://doi.org/10.1109/VISUAL.1997.663897.

Weiskopf D. 2009. Iterative twofold line integral convolution for texture-based vector field visualization. *In*: Möller T, Hamann B, Russell R D eds. Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration. Springer, Heidelberg, Berlin. p.191-211.

Zheng Y H, Ma K, Wang S F, Sun J. 2018. Line integral convolution-based non-local structure tensor. *International Journal of Computational Science and Engineering*, **16**(1): 98-105, https://doi.org/10.1504/IJCSE.2018.089601.