

ELIC and Streamline Injection Based High Quality Visualization for Time-Dependent Surface Flow Field

Huai-Hui Wang, Xun Cai, Wen-Ke Wang, Si-Kun Li
 School of Computer
 National University of Defense Technology, NUDT
 Changsha, China
 huaihuiwang@gmail.com

Abstract—IBFVS is a classical method for visualizing time-dependent surface flow field, but the quality of the final image is inadequate to get a fully understanding for the visualized flow field. In order to improve the quality of the result image of IBFVS, this paper presents an enhanced IBFVS method based on short ELIC filtering and streamline fragments injection. We take IBFVS as the basic mechanism of our method, and use ELIC filtering and streamline injection to process the injected background image to increase the contrast of the result image. Experiment results show that our method generates a better visualization result than IBFVS and other improved methods.

Keywords—component; IBFVS; enhanced line integral convolution; streamline; visualization; time-dependent flow field;

I. INTRODUCTION

Scientific visualization is very important in scientific research, and flow field visualization plays a central role in scientific visualization which is widely used in aircraft design, climate modeling and so on. We know that the data generated from scientific research are so large to analyze straightly, and some effective method such as visualization could be used to make these analyses quite easy and reveal the most important information. Among those visualization methods, texture-based visualization is a promising method compared with others due to its global representation of the whole dataset.

IBFVS (Image based flow visualization for curved surfaces) is an efficient texture-based visualization method for visualizing time-dependent surface vector field, using which noise texture can flow smoothly along the direction of the vector. Although we can dynamically represent flow field with IBFVS, the final image is not clear enough to reveal the fully information carried by the flow field. To solve this inherent disadvantage, researchers use short LIC filter to increase the spatial coherence of the pixels along streamline, and this makes the lines along the streamline clearer. However, this method will decrease the high frequency of the noise, and consequently make the noise image move unnaturally.

In this paper, we propose an enhanced IBFVS method based on short ELIC filtering and streamline fragment injection, which will obviously increase the spatial coherence along streamline and obtain a clear and smooth animation.

II. RELATED WORK

There are many methods to visualize flow field. Direct visualization, such as drawing arrows, is difficult to reveal the continuous flow. Geometric visualization, such as streamline, is integration-based approach which provides more insight, but also cannot show the whole information of flow because of its discreteness. In order to avoid information lost, researchers have spent much effort in developing dense or texture-based visualization methods.

The first attempt of texture-based visualization method is spot noise [1], introduced by Van Wijk. Cabral and Leedom used Line Integral Convolution (LIC) [2], one of the most important approaches of texture-based visualization, to visualize 2D steady flow field, and gives a much higher image quality. Several extensions of LIC are then developed by many researchers such as UFLIC [3], FLIC [4], EFLIC [13] and so on. And then a method using moving textures was introduced to the visualization community [5], which can dynamically visualize 2D flow field. Based on this work, 2D Lagrangian-Eulerian Advection (LEA) [6] was developed, which can produces animations with high spatio-temporal correlation at interactive rates. And then Image-Based Flow Visualization (IBFV) [7] was developed by researchers, which achieves a high performance by using standard features of graphics hardware.

2D Texture-based methods can be extended to 3D vector field. 3D IBFV [8] was presented by Telea et al., which produces sparse 3D noise pattern advectons, thus addresses the occlusion problem inherent to 3D flow visualization. A method for 3D unsteady vector field visualization using real-time advection and volumetric illumination was developed by Weiskopf et al. [9][10]. Falk et al. proposed an output-sensitive 3D line integral convolution method to visualize 3D vector field [11]. All these 3D texture-based method utilized feature detectors to reduce uninterested flow regions and highlight important flow areas.

IBFV was extended to visualize curved surface flow, which is known as IBFVS [12], which produces the texture directly in image space, and uses two-dimensional animated textures to generate surface flow.

III. METHOD

In this section we start with a brief introduction of IBFV for curved surface, and then the new visualization method for

curved surface flow is described, followed by a discussion in more detail.

A. IBFV for curved surface

IBFV for curved surface is developed based on IBFV, which projects the curved surface flow to image space and advects the noise image in 2D image space to generate a surface flow animation. The main ideas of IBFV and IBFV for curved surface will be briefly introduced in this section.

Suppose an unsteady 2D vector field $\mathbf{v}(\mathbf{x};t)$, a pathline is the path of a particle, flows along the direction of the vector, which is the solution of the differential equation[12]

$$\frac{dp(t)}{dt} = v(p(t);t) \quad (1)$$

Where $x \in R^2$ is the position vector of point x , $v(\mathbf{x};t)$ is the velocity of point x at time t , $\mathbf{p}(t)$ is the pathline of a particle.

For a given start point $\mathbf{p}(0)$, we can obtain the first order Euler integration.

$$p_{k+1} = p_k + v(p_k; t)\Delta t \quad (2)$$

Where p_{k+1} is the position of p_k after Δt time interval advection.

Consider a field $F(\mathbf{x};k)$ that represents a series of images advected by the flow, $G(\mathbf{x},k)$ is the injection background image, and α is the blending factor of the advected image and the background image. We can give the output image of each frame.

$$F(p_k; k) = (1 - \alpha)F(p_{k-1}; k-1) + \alpha G(p_k; k) \quad (3)$$

For curved surface flow, the vector is just projected to image space, and the image is advected and injected in image space.

$$F_i(p'_k; k) = (1 - \beta)F_i(p'_{k-1}; k-1) + \beta G(p'_k; k) \quad (4)$$

Where $F_i(x; k)$ is the 2D texture advected in image space, and p'_k is the projected point of p_k from curved surface to image space.

In order to reveal the shape of the curved surface, a shaded image F_s must be blended with the advected image.

$$F(x; k) = \gamma F_i(x'; k) + (1 - \gamma)F_s(x'; k) \quad (5)$$

This is a simple course of visualizing curved surface flow. First, project the vector flow field to 2D space; second, 2D noise image is advected and rendered; third, fresh noise is injected to the image, avoid the disappearance of the noise; fourth, blend the output image with the shaded image of the

curved surface; and finally render the final images to the screen.

B. EIBFV for curved surface

We now consider the improved method, Enhanced IBFV for curved surface. Two key steps are designed to increase the spatial coherence along streamline, ELIC filtering and streamline fragment injection.

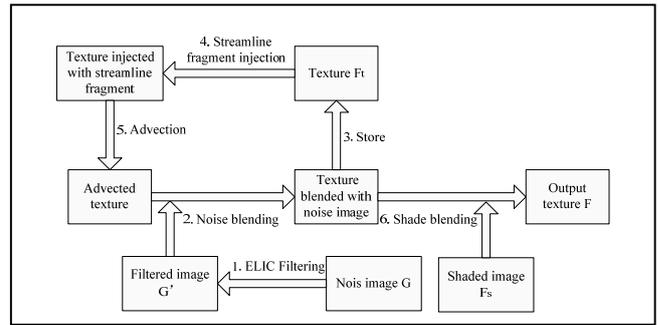


Figure 1. Pipeline of IBFVS.

The main steps of EIBFVS are shown in Fig. 1, in which step 2, 5, 6 are the key steps of IBFVS, and step 1, 4 are the improving processes.

When the noise image G is initialized, step 1 is executed to get a filtered image G' with low frequency in the same streamline and high frequency among different streamlines, and we will discuss ELIC filtering in next part. In step 2, based on equation (4), the noise image will be blended with the advected texture of the last frame which is stored after noise blending and injected with short streamline fragment. Then the result texture will be stored as F_t and blended with shaded image F_s based on equation (5), which could reveal the shape of curved surface, and the result of blending will be copied to the screen as the current output of animation. We will inject some streamline fragment to the stored texture F_t , which will obviously make the streamline-like streak clearer. Then the texture will be advected based on equation (2). Here we use texture coordinates advection in the image space, that is to say, we use the texture coordinate $T_{p_{k-1}}$ of point p_{k-1} as the texture coordinate of p_k .

Consequently, the algorithm of EIBFVS could be formulized as follows.

- a) Generate noise image G ;
- b) Do short ELIC filtering on G and get G' ;
- c) Blend the result image of h) with image G' ;
- d) Store the texture;
- e) Blend the shaded image and render it to the screen;
- f) Inject short streamline fragment to F_t ;
- g) Integrate from current points and find out the texture coordinates of in the late time step;
- h) Render the curved surface with texture F_t and texture coordinates;
- i) Go to step b);

C. Short ELIC filtering

ELIC is known as a two-pass LIC and a high pass filtering [13], shown as follows.

Line Integral Convolution (LIC) can be formulated as follows.

$$F_{out}(x, y) = \frac{\sum_{i=0}^l F_{in}(P_i)h_i + \sum_{i=0}^{i'} F_{in}(P_i')h_i'}{\sum_{i=0}^l h_i + \sum_{i=0}^{i'} h_i'} \quad (6)$$

Where $F_{in}(P_i)$ is the texture at point P_i , i and i' are the convolution length along the positive and negative direction, P_i and P_i' is the i th point along streamline. h_i and h_i' are the weight of texture, integrated from low pass filtering kernel function.

The output of LIC is low frequency image, which can make the color of the image more and more uniformly distributed. A Laplacian high-pass filter is applied to increase the contrast between every pixel. Here we use the following operator.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (7)$$

This operator means the highlight of the central pixel. We use nine times the central pixel color minus its eight neighbor pixels, and the result pixel color is show as the central pixel color.

After the Laplacian high-pass filtering, another LIC should be applied. The pipeline of short ELIC filter is shown as follows.

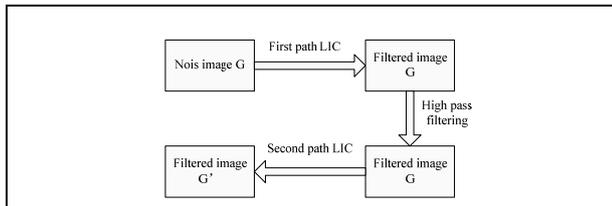


Figure 2. pipeline of short ELIC filter

In Fig. 2, the first pass of LIC will make the color of pixels along the same streamline similar to each other, and then complement high frequency components of the image in order to avoid the increase of the pixels with similar color. The second pass LIC of will lead high quality streamline streaks.

D. Streamline fragment injection

Although short ELIC filter will lead a series of high quality images with clear streamline streaks, the animation produced from these images still flows unnaturally. This is because the high coherence of the streamline streaks between images G' of different phase. In order to solve this problem, some random streamline streaks are injected to texture F_i .

Notice that there are many streamline streaks in F_t . We just need to sharpen them and offer them a random color.

$$C(P_i) = \begin{cases} C_{ran} & N_{ran} \leq C(P_i) \leq N_{ran} + \Delta C \\ C(P_i) & otherwise \end{cases} \quad (8)$$

The sharpen process is formulated by equation (8), where $C(P_i)$ is the result pixel color, C_{ran} is a random color, N_{ran} is random initial value.

The above process will give the streamline streaks with similar color a random color, and just like a streamline fragment injection.

E. Implementation

Texture processing is suitable for graphics hardware implementation, so we implemented the whole algorithm in GPU, especially ELIC, texture advection and streamline fragment injection.

IV. RESULTS AND ANALYSIS

In this section, we will process several curved flow field with IBFVS, IBFVS with LIC filter and EIBFVS. We start with a flow field of a vector dataset of typhoon, and then the average wind stress is visualized.

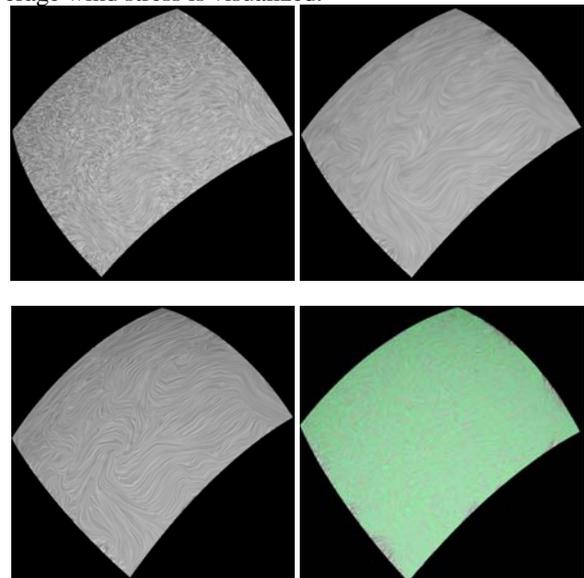


Figure 3. Visualization of isosurface with equal pressure of typhoon.

The visualization results shown in Fig. 3 and Fig. 4 illustrate that image generated with IBFVS is not clear enough to get a fully understanding of the flow field. IBFVS with LIC filter will make the pixels have similar color; IBFVS with ELIC filter and streamline fragments injection can produce a better image with high quality.

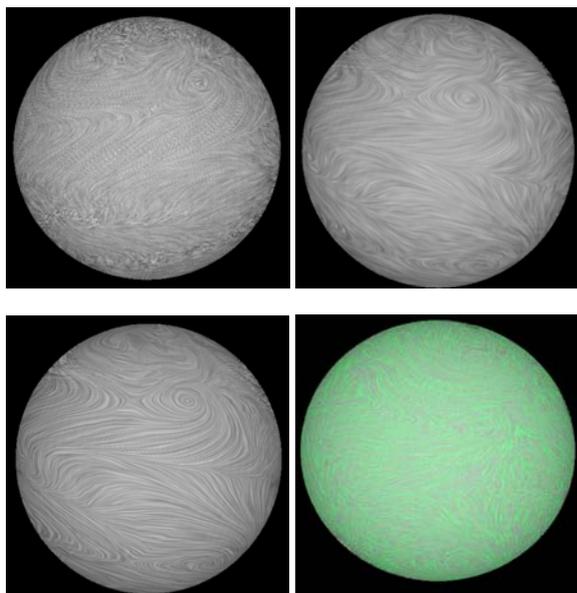


Figure 4. Visualization of average wind stress of the Earth.

V. CONCLUSION AND FUTURE WORK

This paper presents a high quality visualization method for 2D unsteady surface flow field based on IBFV, short ELIC filtering and streamline fragment injection. Experiments illustrate that this method leads a better representation of curved surface flow, and the streakline is highlighted by color de fragments.

In future work, a feature-based visualization should be developed in complicated surface flow, and the efficiency of this algorithm can be enhanced when there is no interaction. Finally, we will extend this high quality method to 3D unsteady flow.

ACKNOWLEDGMENT

This work is financially supported by the National Grand Fundamental Research Program of China (Grant Nos.G 2009CB723803, Grant Nos.G2011AA7123022), the National Natural Science Foundation of China (Grant Nos.

61170157), and the National Natural Science Foundation of China (Grant Nos. 61272009). Thanks for the support of visualization team of Room 623, School of Computer, National University of Defense Technology.

REFERENCES

- [1] J.J. van Wijk (1991). Spot Noise-Texture Synthesis for Data Visualization. *Computer Graphics (Proc. SIGGRAPH '91)*, Volume 25, pp. 309- 318.
- [2] B. Cabral and L.C. Leedom (1993). Imaging Vector Fields Using Line Integral Convolution. In *Proc. ACM SIGGRAPH '93*, pp. 263-270.
- [3] Han-Wei Shen and David L. Kao (1998). A New Line Integral Convolution Algorithm for Visualizing Time-Varying Flow Filed. *IEEE Trans. Visualization and Computer Graphics*, Volume 4, No. 2, pp. 98-108.
- [4] D. Stalling and H. Hege (1995). Fast and Resolution Independent Line Integral Convolution. *Proceedings of ACM SIGGRAPH95, Annual Conference Series*, pp. 249-256.
- [5] N. Max and B. Becker (1995). Flow Visualization Using Moving Textures. In *Proceeding of ICASW/LaRC Symp. Visualizing Time-Varying Data*, pp. 77-87.
- [6] B. Jobard, G. Erlebacher and M.Y. Hussaini (2002). Lagrangian-Eulerian Advection of Noise and Dye Textures for Unsteady Flow Visualization. *IEEE Trans. Visualization and Computer Graphics*, Volume 8, No. 3, pp. 211-222.
- [7] J.J. van Wijk (2002). Image Based Flow Visualization. *ACM Trans. Graphics*, Volume 21, No. 3, pp. 745-754.
- [8] A. Telea and J.J. van Wijk (2003). 3D IBFV: Hardware-Accelerated 3D Flow Visualization. In *Proc. IEEE Visualization*, pp. 233-240.
- [9] D. Weiskopf, T. Schafhitzel and T. Ertl (2005). Real-Time Advection and Volumetric Illumination for the Visualization of 3D Unsteady Flow. In *Proc. EG/IEEE VGTC Symp. Visualization (Eurovis '05)*, pp. 13-20.
- [10] D. Weiskopf, T. Schafhitzel and T. Ertl (2007). Texture-Based Visualization of Unsteady 3D Flow by Real-Time Advection and Volumetric Illumination. *IEEE Trans. Visualization and Computer Graphics*, Volume 13, No. 3, pp. 569-582.
- [11] M. Falk and D. Weiskopf (2008). Output-Sensitive 3D Line Integral Convolution. *IEEE Transactions on visualization and computer graphics*, Volume 14, No. 4, pp. 820-834.
- [12] J. J. van Wijk (2003). Image Based Flow Visualization for Curved Surfaces. *Proceedings IEEE Visualization '03*, pp. 123-130.
- [13] H. Hege and D. Stalling (1998). Fast LIC with Piecewise Polynomial Filter Kernels. *Mathematical Visualization*, pp. 295-314.