Animating Streamlines with Repeated Asymmetric Patterns for Steady Flow Visualization

Chih-Kuo Yeh[*], Zhanping Liu[§ξ] and Tong-Yee Lee[*]

[*]Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, R.O.C.
[§]School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA
[ξ]Division of Computer Science, Kentucky State University, Frankfort, KY 40601, USA

ABSTRACT

Animation provides intuitive cueing for revealing essential spatial-temporal features of data in scientific visualization. This paper explores the design of Repeated Asymmetric Patterns (RAPs) in animating evenly-spaced color-mapped streamlines for dense accurate visualization of complex steady flows. We present a smooth cyclic variable-speed RAP animation model that performs velocity (magnitude) integral luminance transition on streamlines. This model is extended with inter-streamline synchronization in luminance varying along the tangential direction to emulate orthogonal advancing waves from a geometry-based flow representation, and then with evenly-spaced hue differing in the orthogonal direction to construct tangential flow streaks. To weave these two mutually dual sets of patterns, we propose an energy-decreasing strategy that adopts an iterative yet efficient procedure for determining the luminance phase and hue of each streamline in HSL color space. We also employ adaptive luminance interleaving in the direction perpendicular to the flow to increase the contrast between streamlines.

Keywords: Flow visualization, evenly-spaced streamlines, repeated asymmetric patterns, animation, color map

1. INTRODUCTION

There has been significant research on flow visualization including geometry-/glyph-based methods[1] such as arrow plots, streamlines[2,3,4] and stream surfaces as well as texture-/image-based techniques[5], e.g., Line Integral Convolution (LIC)[6] and LEA[7]. Animation is not only an instrumental tool for enhancing the display of spatial correlation such as flow direction, orientation (positive flow direction), and topology but also an intuitive cue for improving the understanding of temporal characteristics like velocity magnitude and flow evolution (e.g., in terms of convergence/divergence) over time. The dynamic nature of flow indicates the intrinsic advantages of animation in conveying the spatial-temporal behavior, particularly for interactive interpretation of complex structures.

There are several challenges for 2D flow animation. ***Spatial continuity***, favored in both tangential and orthogonal directions, facilitates mental reconstruction of the flow even from a geometric representation[2,3,4]. Insufficient continuity may prevent salient features from being recognized since an instantaneous frame leaves less time than provided by a still image for accurate comprehension. ***High image contrast***, e.g., the differentiation between streamlines, helps with the delineation of tangential flow directions for efficient data assimilation. Noisy or blurred images may degrade the power of animation. In fact, image contrast usually involves the use of color as the latter can have a great influence on the former. ***Temporal coherence***, imposed on texture synthesis[7] and layout of geometric elements[4], allows smooth inter-frame transition to be established from visual retention. Failure to address this issue incurs cluttering and flickering problems, affecting perception of the underlying flow structure. Specifically for steady flows, replay jerkiness needs to be solved to produce an elegant cyclic animation. ***Variable-speed*** depiction offers further insight into the flow besides

Further author information: (Send correspondence to Chih-Kuo Yeh)
Chih-Kuo Yeh: E-mail: simpson.ycg@gmail.com
Zhanping Liu: E-mail: zhanpingliu@hotmail.com
Tong-Yee Lee: E-mail: tonylee@mail.ncku.edu.tw

directional information because velocity magnitude relates to such properties as momentum and density. ***The pattern of orthogonal waves***[8] exploits spatial correlation in the direction perpendicular to the flow to add realistic propagating effects. This form of visual stimuli serves as a distinct dual reference, aiding in the sense of the tangential flow movement.

Recently, Chi et al[9] proposed a Self-Animating Image (SAI) technique by which Repeated Asymmetric Patterns (RAPs) are placed along streamlines to generate illusory motion using a single image. The preliminary results demonstrate its value for illustrative flow visualization. However, the size of RAPs usually needs to be large enough to invoke motion illusion in the form of stream ribbons. This restriction hinders their applicability to capturing fine flow features. In this paper, we present an *RAP-based Streamlines Animation* (RAPSA) algorithm by adapting RAPs in terms of size, luminance, and hue with a goal of producing multiple successive images of strong spatial-temporal correlation to compose a high-quality animation of thin streamlines for practical visualization of steady flows. RAPSA addresses all of the aforementioned issues concerning 2D flow animation. Dense evenly-spaced streamlines are used as an aesthetic integral representation that facilitates visual interpolation both along and across the flow to accomplish spatial continuity. This straightforward geometric representation equipped with novel luminance and hue encoding can attain high image contrast. RAPSA employs a unique animation model to achieve tight temporal coherence and variable-speed depiction. In particular, our method is capable of emulating orthogonal waves to strengthen the perception of the tangential flow motion.

The major contributions of this paper lie in the following components. We extend RAPs from illustrative motion interpretation[9] to accurate flow visualization such that they can be applied to dense evenly-spaced streamlines for exploring complex structures. We present a smooth cyclic variable-speed animation model, which utilizes velocity integral luminance transition to encode many levels of visually distinguishable flow magnitude. We propose an energy decrement strategy to fulfill inter-streamline synchronization in luminance varying along the tangential direction, the first approach for building orthogonal advancing waves from a geometric flow representation. We adopt evenly-spaced hue differing followed by adaptive luminance interleaving in the orthogonal direction to accentuate tangential flow streaks for high image contrast.

2. RELATED WORK

There have been a variety of animation techniques for flow visualization, which may be roughly categorized as follows. ***Geometry succession*** resorts to progressive change in size/length, position, and/or shape of geometric objects, e.g., massless particles and deformed surfaces, to capture snapshots of the ‘real’ motion. This type of methods is straightforward and usually easy to implement, though care needs to be taken to address spatial-temporal coherence. ***Color-table transfer*** takes luminance transition as an orientation metaphor to map the primitives of the animated flow pattern into an array of intensity-tapering color entries[4]. This family of approaches is good at maintaining temporal coherence and well suited for visualizing the variable speed by the luminance gradient. ***Phase shifting***, often used in texture-based flow visualization, works by varying the phase of a periodic filter during image synthesis to create a sequence of frames for a cyclic animation[6]. This kind of techniques exploits the local phase change of a stationary pattern to construct the sense of a global ”motion without movement”[10]. However, this mechanism has been acclaimed to be weak in variable-speed flow delineation, encoding very limited ranges of visually discernible velocity magnitude in the resulting animation. ***Frame blending*** is widely applied to texture-based unsteady flow visualization by leveraging hardware (e.g., GPU) capabilities for accelerated implementation of flow (particles) advection over successive frames[7,8]. In general, an additional rendering pass is needed as a post-process to improve image contrast.

To our knowledge, Bachthaler and Weiskopf’s approach[8] is currently the only one in the literature related to our method regarding the ability to emulate orthogonal waves, though the two differ in the following aspects. The former produces advancing strips perpendicular to the flow through a LIC-like texture synthesis process while the latter creates this visual effect by imposing synchronization on dense evenly-spaced streamlines (a geometric representation) in periodic luminance tapering. In fact, the former adopts indirect flow depiction via noise advection and texture convolution, whereas the latter performs direct flow visualization with explicit integral curves color-mapped by RAPs. This shows a major difference between the two in the working principle. As a texture synthesis technique, the former is intrinsically compute-intensive and requires GPU capabilities for hardware acceleration. The latter is GPU-independent and possesses great potential of interactive flow visualization thanks to the recent advances in evenly-spaced streamline placement[2,3]. Thus our method offers an alternative way of creating orthogonal waves to help with the visualization of tangential directions in a dense fashion. Meanwhile, it supports effective variable-speed flow animation, at which some texture-based techniques like LIC are not good.

3. RAP-BASED STREAMLINES ANIMATION

This section describes our RAP-based streamlines animation algorithm, i.e., RAPSA, which employs farthest point seeding[2] to place long evenly-spaced streamlines. By rendering the patches resulting from Delaunay triangulation of the streamline layout, RAPSA ensures full coverage of the flow domain. With a deliberately designed HSL-space color map scheme, RAPSA can show well-contrasted tangential flow streaks and visually appealing orthogonal cascading waves in animated frames.

We begin with a basic RAP animation model that uses velocity integral luminance transition to achieve smooth cyclic variable-speed flow motion. What follows is the extension of this model by synchronizing neighboring streamlines, *wherever possible*, in luminance varying along the tangential flow direction to form orthogonal propagating waves. Then we refine the model further with evenly-spaced hue differing in the orthogonal direction for exposing tangential flow streaks. To combine these two inter-perpendicular families of patterns, an energy-decreasing strategy is presented to obtain the luminance phase and hue of each streamline by rapid iteration. We conclude the pipeline with adaptive luminance interleaving to add to the inter-streamline contrast initially established by hue differing.

3.1 Creating Cyclic Variable-Speed Animation

Intensity tapering is an effective motion metaphor. We adopt TYPE I RAP[11] as the intensity ramp that is placed along streamlines to indicate the flow orientation. To make efficient use of the color spectrum, we choose HSL (hue, saturation and lightness) color space for mapping the RAP lightness (or *luminance*) and the associated hue to streamlines. In this paper, the luminance and hue of point n ($n \in [0, N\text{-}1]$ where N is the number of points/samples of a streamline and point 0 lies in the upstream side, i.e., the *starting point*) on streamline s are denoted $L_s(n)$ and $H_s(n)$, respectively, which are both scaled to [0, 1). We delay the discussion of hue encoding until section 3.3. Then the RAP luminance along a streamline is given by

$$L_s(n) = \mathrm{mod}(\ \theta_s + f \times d_s(n)\ ) \tag{1}$$

$$\mathrm{mod}(x) = x - \lfloor x \rfloor \tag{2}$$

where $\theta_s \in [0, 1)$ is the phase of streamline s and $d_s(n)$ is the curved distance between the starting point and point n along streamline s with $d_s(0) = 0$. Parameter $f > 0$ is the *frequency* at which the luminance varies with the arc length. Function *mod* transforms the virtual, monotonically increasing luminance to [0, 1), i.e., the range of the real valid luminance that bounds any full RAP. With this luminance range, a smaller value of f results in a (spatially) longer RAP and vice versa, while a streamline is decomposed into multiple sequentially connected RAPs (Fig. 1). Given T as the number of successive frames constituting a cyclic animation, the full span of any RAP, regardless of the phase, is then accessed equally by the T frames and the inter-frame shift *in space* (i.e., the distance between any two successive access locations) along the RAP is $1/T$ times the whole ramp size. As the frames are animated, a longer RAP by a smaller f exhibits a greater spatial change and hence a higher speed than a shorter RAP by a larger f does.

A cyclic animation is made up of T frames if and only if the inter-frame shift *in the luminance* of any point (on each streamline present in the scene) accessing a complete RAP is $1/T$. In other words, a steady flow can be animated by cyclically varying the luminance of each point along a streamline by $1/T$. Thus the initial formulation of the luminance of point n on streamline s, given by Equation (1), can be extended below with the addition of the temporal dimension for creating a cyclic animation of T successive frames.

$$L_{st}(n) = \mathrm{mod}\left(L_s(n) - \frac{t}{T} \right) \tag{3}$$

where $t \in [0, \mathrm{T}\text{-}1]$ is an integer indicating the frame index and $L_{st}(n)$ is the spatial-temporal description of the luminance. With parameter f fixed for all points throughout the flow domain, a single RAP is then placed along streamlines, producing an equal-speed animation. By adjusting f based on the point with the smallest value for the highest speed and the largest for the lowest, multiple RAPs of different lengths are mapped to streamlines, yielding a variable-speed animation. Fig. 1 illustrates the use of f for equal-speed (Fig. 1a) and variable-speed (Fig. 1b) animations.

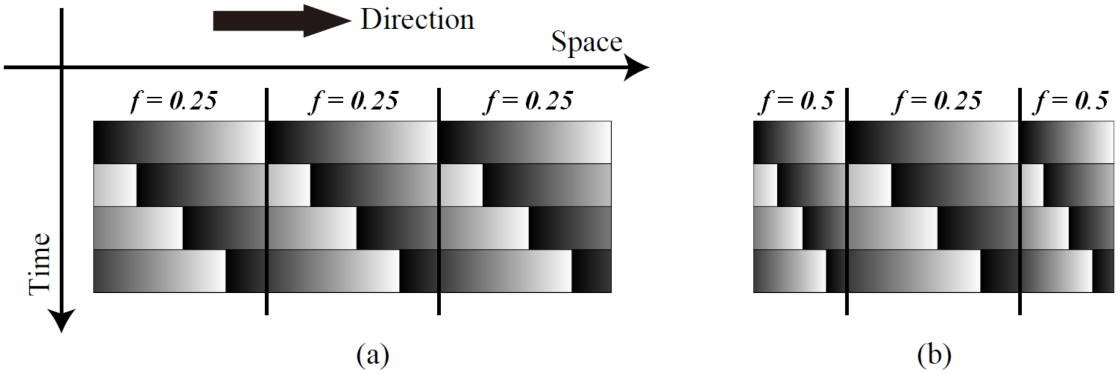

Figure 1. RAPs and the possible values of parameter *f* for generating (a) equal-speed and (b) variable-speed motion effects. With the full span of each RAP accessed uniformly by four successive frames (indexed downward), the longer RAP by the smaller value of *f* (0.25) exhibits a greater inter-frame spatial change and hence a higher speed in the resulting animation than the shorter RAP by the larger value of *f* (0.50) does. Note that the perceived motion direction goes from left to right as the RAP luminance varies from darkness to brightness to create Fraser-Wilcox illusion[11].

Equation (1) employs parameter *f* to encode a pseudo variable-speed effect in animated frames. In fact, the velocity magnitude of a vector field is an optimal choice of data to generate more reasonable variable-speed flow animation. The original velocity magnitude is converted/scaled to range [0, 1], usually by means of a nonlinear transformation such as histogram equalization. Given the post-conversion velocity magnitude $v_s(n)$ at point n on streamline s, the luminance $L_s(n)$ is computed by

$$\delta_s(n) = \tan\left(\beta_{\max} - v_s(n) \times \left(\beta_{\max} - \beta_{\min}\right)\right) \tag{4}$$

$$\ell_s(n) = \sum_{i=1}^{n} \delta_s(i) \times \left(d_s(i) - d_s(i-1)\right) \text{and } \ell_s(0) = 0 \tag{5}$$

$$L_s(n) = \operatorname{mod}\left(\theta_s + \ell_s(n)\right) \tag{6}$$

where β_{min} and β_{max}, $0<\beta_{min}<\beta_{max}<0.5\pi$, are two threshold angles used in a monotonically decreasing function $\delta_s(n)$ to map the lowest velocity magnitude to $\tan(\beta_{max})$ and the highest to $\tan(\beta_{min})$. $\delta_s(n)$, a point-varying/-based representation of the frequency *f* used in Equation (1), governs the amount of local change in luminance per unit arc length at point n on streamline s. As an approximation to the accumulated luminance change across infinitesimal arc segments from the starting point till point n, $\ell_s(n)$ delineates the *velocity* (or magnitude-driven) *integral luminance transition* along streamline s. This improved alternative to the latter part ($f \times d_s(n)$) of Equation (1), in combination with phase θ_s, determines $L_s(n)$ by indexing the RAP ramp.

3.2 Emulating Orthogonal Cascading Waves

Latest research[8] shows that orthogonal waves not only produce realistic visual effects but more importantly augment the comprehension of the tangential flow motion. To emulate highly noticeable advancing ripples, spatial-temporal correlation needs to be imposed on $L_{st}(n)$ between neighboring streamlines, as opposed to applying separate RAP map texturing to each streamline (Fig. 2a). According to Equation (3), it suffices to establish spatial correlation between streamlines in $L_s(n)$. We fulfill this constraint by synchronizing streamlines in the orthogonal direction as they undergo luminance transition along the tangential flow direction. It is worth noting that the degree of the ultimate synchronization depends on the relationship between neighboring streamlines in both flow convergence/divergence and velocity magnitude.

This s*ynchronized luminance transition* scheme may be reformulated as decreasing the following $E(L)$ function

$$\Delta(\Phi, L, s, n) = \operatorname{mod}\left(\ \Phi\left(L_s(n)\right) - L_s(n) + 0.5\ \right) - 0.5 \tag{7}$$

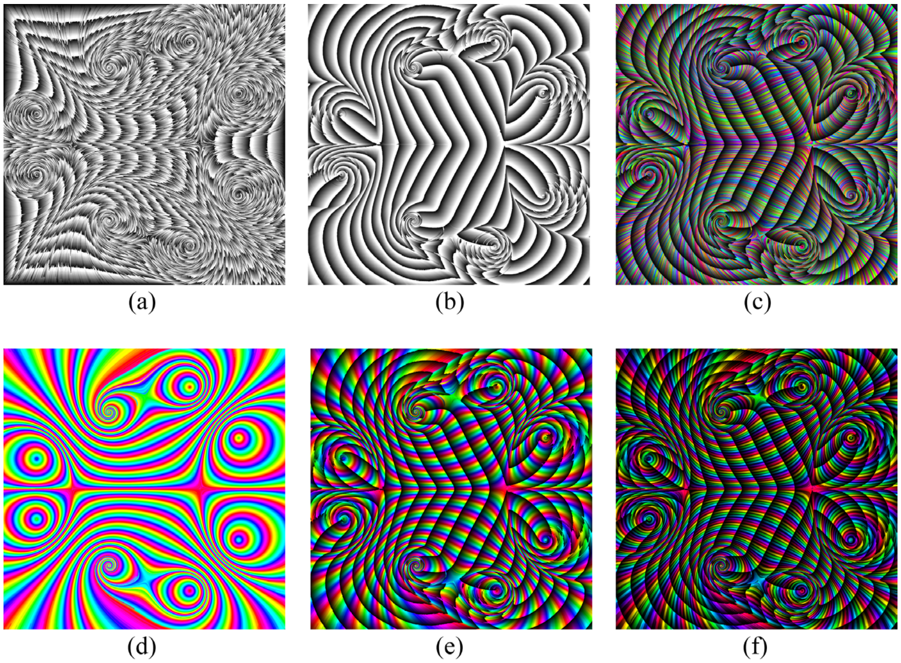

Figure 2. Emulation of orthogonal cascading waves and construction of tangential flow patterns. (a) As streamlines are separately mapped with the RAP luminance, RAP segments overlap one another to form an unorganized pattern. This layout not only fails to build clear orthogonal strips but also affects the comprehension of tangential flow directions. (b) With streamlines synchronized in luminance transition, neat orthogonal advancing waves are synthesized. Without hue encoding, this mechanism alone eliminates flow streaks, sacrificing the intended functionality. (c) The lack of appropriate control over inter-streamline hue differing may cause excessive contrast, degrading the image with distracting artifacts. (d) Under energy control, evenly-spaced yet smooth hue differing between streamlines can be achieved to convey tangential flow directions. (e) The combination of orthogonal cascading waves with tangential flow patterns leads to effective visualization of the flow. However, the visual granularity provided by the hue differing mechanism hinders high-resolution investigation of tangential flow directions. (f) The addition of hue-driven luminance interleaving in the orthogonal flow direction reveals more streamlines to improve the image contrast without incurring artifacts.

$$\Delta(L,\Psi,s,n)=\mathrm{mod}(\ L_s(n)-\Psi(L_s(n))+0.5\ )-0.5 \tag{8}$$

$$E(L)=\frac{1}{2}\sum_{s=0}^{S-1}\sum_{n=0}^{N-1}\left\|\Delta(\Phi,L,s,n)-\Delta(L,\Psi,s,n)\right\|^2 \tag{9}$$

where $\Phi(L_s(n))$ and $\Psi(L_s(n))$ are the luminance values of the two *o*rthogonally *i*mmediate *n*eighbors (OINs) of streamline s at point n, respectively. Thus $\Delta(\Phi,L,s,n)$ and $\Delta(L,\Psi,s,n)$ each denote the difference in luminance between streamline s and an OIN at point n. The energy term $E(L)$ is a metric of the overall inter-streamline luminance discontinuity. S is the number of streamlines and N is the number of points/samples of each. The decrement of $E(L)$ causes neighboring parallel streamlines with similar speeds along the flow to take approximate luminance values in a synchronous manner, creating propagating strips across the flow (Fig. 2b).

3.3 Constructing Tangential Flow Patterns

In sections 3.1 and 3.2 the hue of point n on streamline s, i.e., $H_s(n)$, is temporarily disregarded and therefore orthogonal waves just smear out streamlines, causing poor image contrast or lack of inter-streamline differentiability (Fig. 2b). Now we exploit hue encoding to construct tangential flow directions over orthogonal cascading ripples to weave these two mutually dual sets of patterns. With the hue fixed along each streamline but varying between streamlines, this visual identity allows individual flow streaks to be recognized. While hue differing needs to be sufficient for inter-streamline differentiation, it must be under appropriate control to avoid excessive hue contrast that may introduce artifacts to the image (Fig. 2c) and flickering to the animation. Uniform hue transition in the orthogonal flow direction, facilitated by evenly-spaced streamlines, is a natural solution for meeting these two requirements.

Evenly-spaced hue differing needs to be governed by thresholding the following *E*(*H*) function

$$\Delta(\Phi, H, s, n) = \mathrm{mod}(\Phi(H_s(n)) - H_s(n) + 0.5) - 0.5 \tag{10}$$

$$\Delta(H, \Psi, s, n) = \mathrm{mod}(H_s(n) - \Psi(H_s(n)) + 0.5) - 0.5 \tag{11}$$

$$E(H) = \frac{1}{2}\sum_{s=0}^{S-1}\sum_{n=0}^{N-1}\left\|\Delta(\Phi, H, s, n) - \Delta(H, \Psi, s, n)\right\|^2 \tag{12}$$

where $\Phi(H_s(n))$ and $\Psi(H_s(n))$ are the hue values of the two OINs of streamline *s* at point *n*, respectively. $\Delta(\Phi, H, s, n)$ and $\Delta(H, \Psi, s, n)$ each denote the difference in hue between streamline *s* and an OIN at point *n*. The energy term *E*(*H*) is a metric of the overall inter-streamline hue discontinuity. Thresholding *E*(*H*) to an appropriate range [e_{h0}, e_{h1}] steers evenly-spaced hue transition between streamlines to accentuate tangential flow directions while suppressing artifacts in the image. If an implementation begins with high-energy evenly-spaced hue differing, as is the case with ours (section 3.4), energy thresholding just turns into an energy decreasing process.

3.4 Determining Phase and Hue

We have presented two energy functions, one for synchronized luminance transition to emulate orthogonal cascading waves (section 3.2) and the other for evenly-spaced hue differing to construct tangential flow patterns (section 3.3). The fundamental task is to determine the phase θ_s and hue H_s for each of the evenly-spaced streamlines present in an image. These two streamline attributes can be obtained via iterative equations

$$\theta_s^{m+1} = \theta_s^m + \frac{1}{N}\sum_{n=0}^{N-1}\frac{(\Delta(\Phi, L, s, n) - \Delta(L, \Psi, s, n))}{2} \tag{13}$$

$$H_s^{m+1} = H_s^m + \frac{1}{N}\sum_{n=0}^{N-1}\frac{(\Delta(\Phi, H, s, n) - \Delta(H, \Psi, s, n))}{2} \tag{14}$$

where *m* is the iteration index and as it increases, the two energy functions decrease (as discussed in Section 3.2, flow divergence/convergence and location-based speed variance may keep *E*(*L*) from decreasing after some number of iterations). In essence, the iteration updates the attribute (θ or *H*) of each streamline based on its difference from the average of the two OIN streamlines (see the definitions given by Equations 7, 8, 10, and 11). This procedure smoothens the gap between streamlines in the phase and hue, reducing the overall energy.

Our iterative implementation can be accelerated by an optimal initialization of the two attributes — θ_s^0 and H_s^0. The basic idea is to find an as-long-as-possible (orthogonal) curve that is perpendicular to some evenly-spaced streamlines, of which the as-middle-as-possible one yet with two known attributes serves as a reference to initialize the other streamlines in a bi-directionally spreading manner. Given a hue interval *h*, the streamlines are assigned with evenly-spaced hue values while the phase of each streamline is determined by the luminance difference from the preceding streamline. To perform initialization on all tangential streamlines, *orthogonal streamlines* — evenly spaced in the orthogonal vector field (generated by rotating the original vector field by 90 degrees) with the same density as that of the tangential streamlines, are retrieved for the aforementioned longest spreading paths. The pseudo code for initializing the phase and hue values is given in Fig. 3.

Fig. 2d demonstrates the sole use of smooth evenly-spaced hue differing for constructing tangential flow patterns. Fig. 2e shows the result of combining these tangential flow patterns with the orthogonal cascading waves created by synchronized luminance transition.

3.5 Enhancing Contrast Between Streamlines

Given an inter-streamline hue interval *h* (section 3.4) that is small enough to suppress distracting artifacts, the visual granularity tends to lead to insufficient image contrast (Fig. 2d and Fig. 2e). This dilemma indicates the restriction of the mere use of hue differing for accentuating streamlines. To address this issue, we propose to apply adaptive luminance interleaving, in the orthogonal flow direction, to the result of hue differing. The initial hue-based inter-streamline resolution is increased by the addition of narrow zebra ribbons as a visual sub-division of hue bands. With the hue values unchanged, an appropriate alteration to the luminance values within a hue-guided sub-range in the orthogonal direction to make dimmer zebra ribbons can enhance the contrast between streamlines without introducing artifacts. To implement adaptive luminance interleaving, the original luminance $L_s(n)$ is adjusted with the hue H_s by

$$L_s(n) = \begin{cases} L_s(n) & \text{if } \mathrm{mod}(\gamma \cdot H_s) < 0.5 \\ \frac{1}{2}\left(L_S(n)\right)^2 & \text{if } \mathrm{mod}(\gamma \cdot H_s) \geq 0.5 \end{cases} \tag{15}$$

where γ is a parameter governing the width of a zebra ribbon. The upper part of the equation refers to the hue-low-pass operation band and the lower conducts luminance suppression.

```
TS = set of Tangential Streamlines evenly spaced with placement density $\lambda$;
OS = set of Orthogonal Streamlines evenly spaced with placement density $\lambda$;
STS = queue of Seed Tangential Streamlines $\in$ TS, used to determine orthogonal spreading paths;

foreach ( streamline ts in TS ) ts.initialized = 0;
foreach ( streamline ts in TS )
{   if ( ts.initialized != 0 ) continue;
    ts.$\theta$ = 0;      ts.H = 0;      ts.initialized = 1;   // 1: possibly with further updating
    STS.push( ts );

    while ( STS is not empty )
    {   sts = STS.pop();
        os = the longest orthogonal streamline $\in$OS that intersects with sts;
        OS.erase( os );

        streamline set tsLines = { sts } + { tangential streamlines $\in$ TS that intersect with os };
        streamlines in tsLines are spatially numerated (along os) from 0 to (nLines – 1);
        for ( i = 0; i < nLines; i ++ )
        {   // use linear interpolation to compute the following two values
            lumin[i] = $\ell_{tsLines[i]}$(the intersection between os and tsLines[i]); // equation (5)
            LUMIN[i] = $L_{tsLines[i]}$(the intersection between os and tsLines[i]); // equation (6)
        }
        referenceId = Id of an initialized tangential streamline that is the closest to index nLines/2;
        tsLines[referenceId].initialized = 2;      // 2: no further updating
        for ( i = referenceId; i < nLines – 1; i ++ )
        if ( tsLines[i+1].initialized != 2 )
        {   tsLines[i+1]. $\theta$ = mod( LUMIN[i] – lumin[i+1] );
            tsLines[i+1].H = mod( tsLines[i].H + h ); // h: inter-streamline hue interval
            tsLines[i+1].initialized = 2;      // 2: not further updated
            if ( tsLines[i+1] != ts ) STS.push( tsLines[i+1] );
        }
        for ( i = referenceId; i > 0; i -- )
        if ( tsLines[i-1].initialized != 2 )
        {   tsLines[i-1]. $\theta$ = mod( LUMIN[i] – lumin[i-1] );
            tsLines[i-1].H = mod( tsLines[i].H - h ); // h: inter-streamline hue interval
            tsLines[i-1].initialized = 2;      // 2: not further updated
            if ( tsLines[i-1] != ts ) STS.push( tsLines[i-1] );
        }
    }
}
```

Figure 3. Pseudo code for initializing the phase and hue values of streamlines.

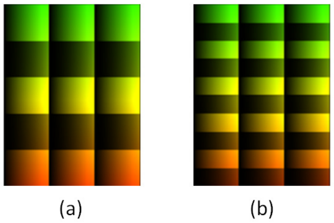

Figure 4. The role of parameter γ in determining the resolution of luminance interleaving on a pattern created by the combination of synchronized luminance transition and evenly-spaced hue differing (with a certain hue interval h) from a horizontal flow. (a) Five zebra ribbons come into view when γ is 10. (b) Ten zebra ribbons are discernible when γ is 20. The bright ribbons result from the hue-low-pass part and the dim from the luminance-suppression part.

Fig. 4 illustrates the effect of parameter γ on the density of luminance interleaving, given a certain hue interval h. As γ gets larger, more streamlines are revealed. In fact, the selection of γ is closely related to the value of the hue interval h since both have direct influences on inter-streamline contrast. They differ in the working mechanism, with one in the luminance component and the other in the hue component. Their multiplication controls the ultimate inter-streamline differentiation. Fig. 5 portrays the correlation between these two parameters in contributing to inter-streamline contrast. Satisfactory results can be obtained by using $h = 0.01$ and $\gamma = 20$ in this paper. Equation (15) may be modified regarding the threshold (0.5) and the luminance suppression function for specific effects.

Fig. 2f shows the result of performing adaptive luminance interleaving after synchronized luminance transition and evenly-spaced hue differing, i.e., the image generated by running through the entire (standard) pipeline of RAPSA. More streamlines are exposed in Fig. 2f than in Fig. 2e, increasing the overall image contrast while avoiding artifacts.

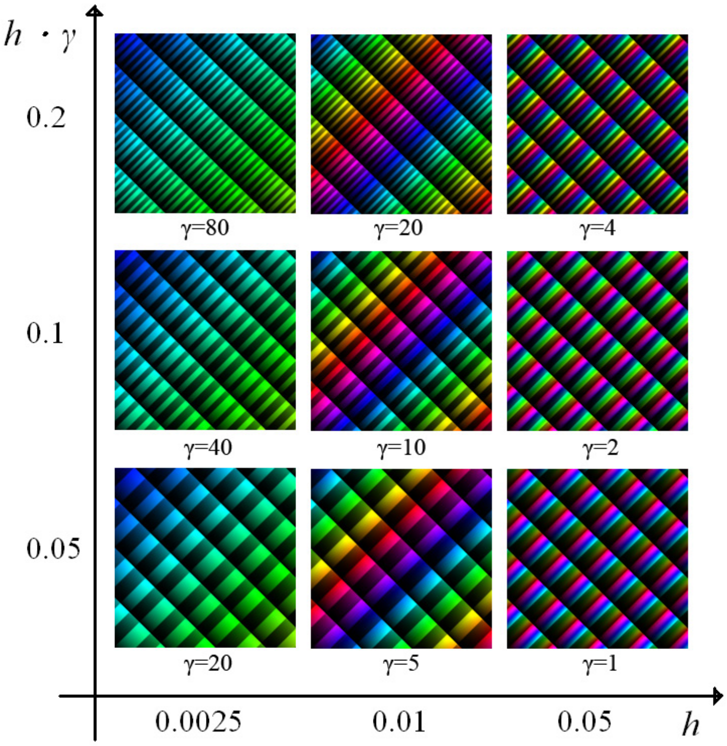

Figure 5. The correlation between the hue interval h (section 3.4) and the luminance interleaving parameter γ (section 3.5) in the overall inter-streamline contrast of the visualization of an anti-diagonal flow. The images along each column (with h fixed) reflect the influence of γ and the images along each row indicate that of various combinations of h and γ (with their multiplication fixed) on differentiating streamlines.

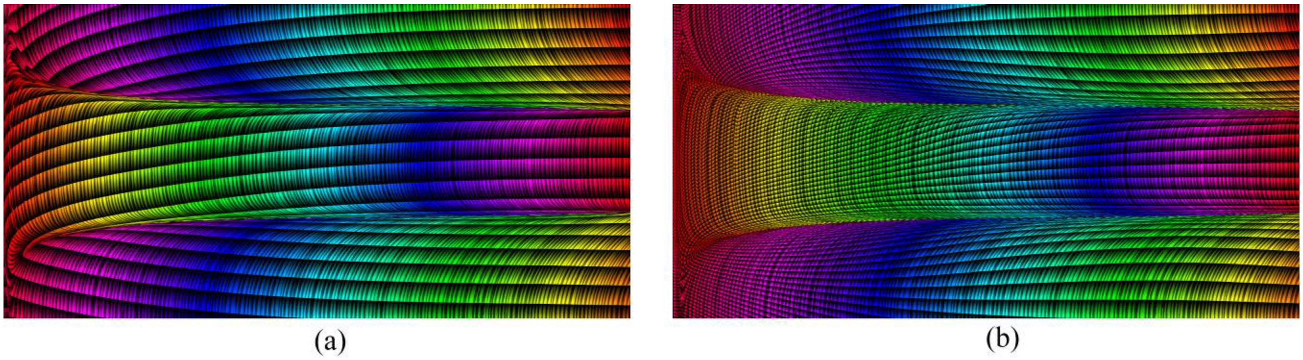

Figure 6. Two RAPSA images generated in (a) constant-speed mode and (b) variable-speed mode, respectively, for visualizing a 770 x 386 flow field. Smooth evenly-spaced hue differing, combined with adaptive luminance interleaving, accentuates individual flow streaks in HSL space to yield high inter-streamline contrast in the two RAPSA images yet without introducing artifacts. The perception of these tangential flow patterns is augmented by the orthogonal cascading waves built through synchronized luminance transition. The length of RAPs, uniform in (a) but non-uniform in (b), encodes the velocity magnitude.

4 RESULTS AND DISCUSSIONS

We have implemented our RAPSA algorithm using Visual C++ on a PC (Intel i7 2.8GHz, 4GB RAM, without GPU support) running Windows Vista. With 1.5% as the streamline layout density for the results reported in this section, the overall cost for producing an animation of 8 frames involves tangential streamline placement[2] (0.40 ~ 0.60 seconds), orthogonal streamline placement (0.30 ~ 0.50 seconds), luminance and hue computation (Section 3, 0.4 ~ 0.6 seconds), Delaunay triangulation (3.00 ~ 4.50 seconds), phase shifting (negligible), and rendering of triangles (negligible). It is worth mentioning that this timing information is based on a preliminary software-based implementation without much performance optimization. The former two components can be improved by using a faster streamline placement method[3]. Although our RAPSA algorithm is not built on GPU, the Delaunay triangulation part (currently the bottleneck of the pipeline) may be significantly accelerated by a GPU implementation[12]. The discussions below are primarily focused on several issues in terms of 2D flow animation quality, i.e., spatial continuity, high image contrast, temporal coherence, variable-speed delineation, and synthesis of orthogonal waves. To demonstrate the spatial-temporal quality of our algorithm, we provide an accompanying animation for each RAPSA image in this section.

Fig. 6 shows two RAPSA images produced in constant-speed mode (the simplified mode, Fig. 6a) and variable-speed mode (the standard mode, Fig. 6b), respectively, for a 770 × 386 flow field. Each image gives a dense flow representation by applying RAPs to high-density evenly-spaced streamlines and then rendering the Delaunay-triangulated color-mapped patches to guarantee full coverage of the flow domain. On the other hand, streamlines still can be easily distinguished from one another without distracting artifacts due to smooth evenly spaced hue differing followed by adaptive luminance interleaving, both in the orthogonal flow direction. In Fig. 6a, uniform-size RAPs are employed for streamlines, resulting in a constant-speed flow animation. Fig. 6b demonstrates the use of variable-size RAPs for creating a variable-speed flow animation, with long luminance-ramps exhibited in rapid flow areas (lower right and upper right), medium in modest areas (middle right), and short in stagnant areas (left). Note that the presence of tiny RAPs in stagnant flow areas does not degrade the image contrast thanks to the effectiveness of evenly-spaced hue differing and adaptive luminance interleaving. Woven with the highly-contrasted streamlines — the tangential flow patterns in both images, are the advancing waves aligned along the curves perpendicular to the flow through synchronized luminance transition. These visually pleasing orthogonal patterns strengthen the sense of the tangential flow patterns, particularly when a sequence of temporally coherent frames is animated. With variable-speed visualization, an RAPSA animation can improve the understanding of the flow dynamics in detail.

Fig. 7 shows six RAPSA images generated by visualizing synthetic 400 × 400 datasets containing complex flow patterns, i.e., two with x-axis symmetry (left column), two with y-axis symmetry (middle column), and two with center-based symmetry (right column). These RAPSA images are good at displaying the tangential flow direction (the local information) and revealing the flow topology (the global structure) and constituent critical points. Evenly-spaced

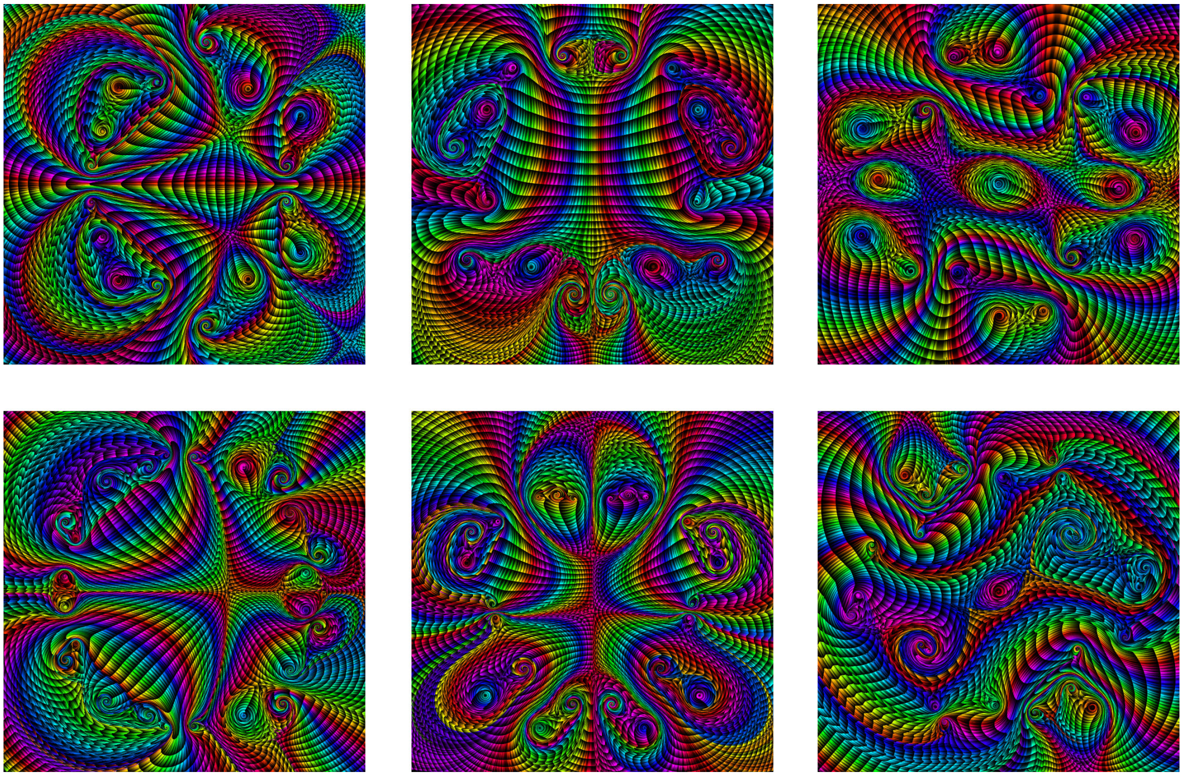

Figure 7. RAPSA images for the visualization of two x-axis symmetric (left column), two y-axis symmetric (middle column), and two center-based symmetric (right column) 400 x 400 complex flow datasets.

hue differing offers great aid in the analysis of the underlying interaction between individual flow elements. As a streamline identity, the hue channel allows a flow field to be 'contoured', helping locate the flow separatrices that decompose the entire domain into multiple regions.

Fig. 8 shows two images generated using SAI[9] and RAPSA, respectively, for a 576 × 291 wind flow field. The SAI image (Fig. 8a) conveys the flow in a discrete manner and can only serve as an illustrative interpretation by the illusionary motion effect. It is cluttered by the glyphs due to their size and the simplistic layout along the flow. The RAPSA image (Fig. 8b) gives a dense flow representration. The curve segments aligned in the cascading strips can be successively connected through the differing hue to visually reconstruct a high-density placement of evenly-spaced streamlines. This individual nature of streamlines ensures high image contrast in the dense representation.

Fig. 9 demonstrates the use of our RAPSA algorithm for visualizing a 600 × 400 flow field, part of a US Navy model of the Northeast Pacific Ocean (143.31° W ~ 110.76° W, 49.63° N ~ 62.00° N), with "degenerate" (Fig. 9a) and standard (Fig. 9b) evenly-spaced hue differing. In the former case, normal hue differing (section 3.4) is performed in support of subsequent luminance interleaving (Equation 15), though a single hue (blue herein) instead of a band of varying hue values is applied to the streamlines in the output image. Even without standard hue mapping, Fig. 9a is still able to exhibit flow streaks and the blueness mimics the real-world ocean apperance. Fig. 9b is focused on the insight into the data and needs less mental efforts for reconstructing the integral lines. With the land (upper right of the domain) textured as the context, both images are capable of displaying the vortical structures and propagating behavior of the flow such as the inshore advancing waves.

Fig. 10 gives three snapshots of animating RAPSA images to visualize a 500 × 500 2D slice of Hurricane Isabel (83.00° W ~ 62.00° W, 23.70° N ~ 41.70° N) at time step 24. A sequence of RAPSA images is repeated for multiple cycles, with one opacity per cycle used to composite the images and the underlying land and sea. Given a set of opacities increasing from 0 (Fig. 10a) to a fraction (Fig. 10b) and further untill 1 (Fig. 10c) and decreasing back to 0 (Fig. 10a), the entire multi-cycle animation associates the flow pattern (e.g., the eye of Isabel at the center) with the physical location.

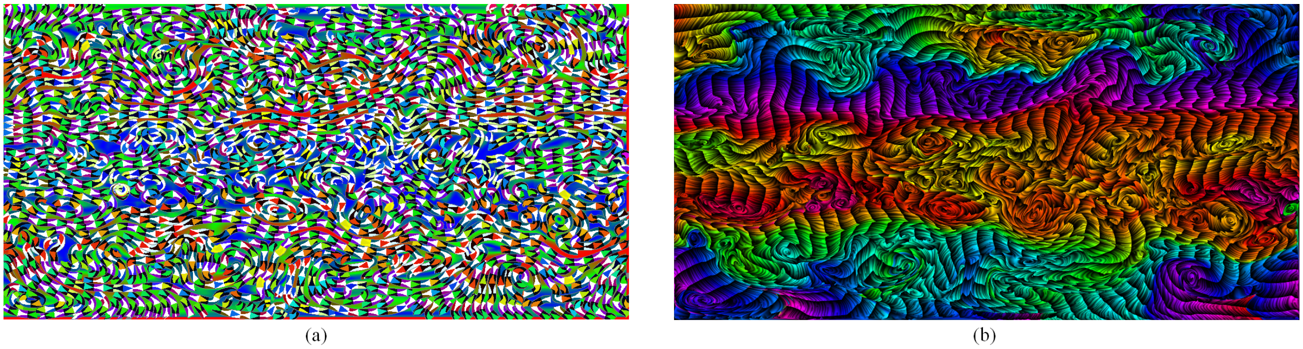

Figure 8. Comparison of SAI[9] and RAPSA in displaying a 576 × 291 wind flow. The SAI image (a) suffers from low spatial resolution and glyph cluttering. The RAPSA image (b) provides a dense representation of the flow in both the tangential/integral direction and the orthogonal direction, while flow streaks are still easily discernible.

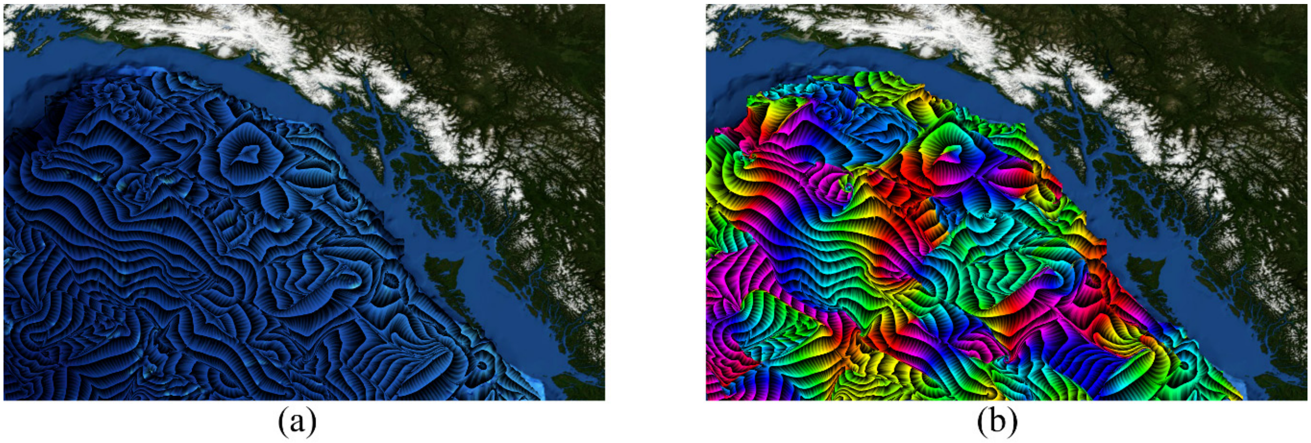

Figure 9. Two RAPSA images for visualizing the Northeast Pacific Ocean flow. Both followed by luminance interleaving, (a) 'degenerate' evenly-spaced hue differing imitates the real ocean flow with blueness and (b) standard hue differiing is intended to emphasize the insight.

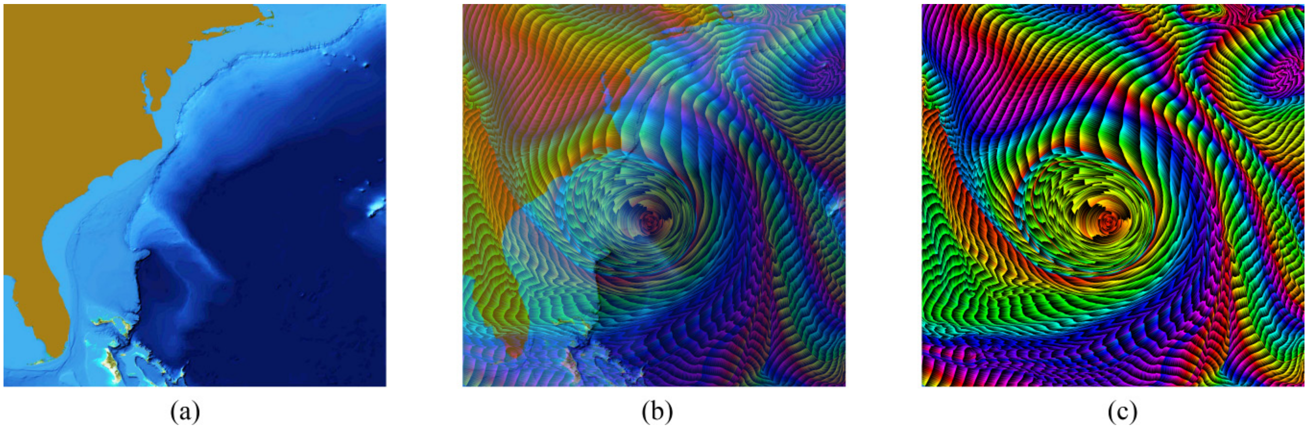

Figure 10. Three snapshots of a multi-cycle animation of RAPSA images, blended with the context through a series of opacities that increase from 0 (a) to a fraction (b) and further until 1 (c) and then decrease back to 0 (a), for visualizing a 2D slice of Hurricane Isabel at time step 24. An individual image captures the instantanesous structure of Isabel and the animation helps explore its development.

Though created for a single time step of the data, this cyclic animation can provide important clues for investigating the trend of hurricane evolution.

5. CONCLUSIONS AND FUTURE WORK

We have presented RAPSA, an RAP-based streamlines animation algorithm that maps repeated asymmetric patterns of deliberately designed spatial-temporal correlation to high-density evenly-spaced integral lines for visualizing steady flows. We propose a smooth cyclic variable-speed flow animation model based on velocity (magnitude) integral luminance transition. Further imposed on this model is inter-streamline synchronization in luminance varying for emulating orthogonal cascading waves, followed by evenly-spaced hue differing between streamlines for constructing tangential flow directions. The two mutually dual sets of patterns are woven together using an efficient iterative implementation of our energy-decreasing strategy. We adopt adaptive luminance interleaving to enhance the contrast in the direction perpendicular to the flow. Results indicate that as a geometry-based method, RAPSA breaks the restriction of previous RAP techniques[9], enabling dense accurate visualization of complex real world flows via animated streamlines in elegant placement coupled with visually appealing orthogonal advancing waves.

As for future work, we may accelerate the algorithm regarding various components of the pipeline discussed in Section 3. More importantly, we would like to extend our approach for time-varying flows and curved surface flows. The former requires some changes to the streamline placement strategy for inter-timestep coherence and the latter involves view-dependency issues. We are also interested in their combination, i.e., visualization of unsteady surface flows.

ACKNOWLEDGEMENTS

This work was supported in part by the Landmark Program of the NCKU Top University Project (contract B0008) and the National Science Council (contracts NSC-99-2221-E-006-066-MY3, NSC-100-2221-E-006-188-MY3 and NSC-100-2628-E-006-031-MY3), Taiwan. The authors would like to thank the U.S. National Center for Atmospheric Research for the hurricane Isabel dataset.

REFERENCES

[1] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen, “Over Two Decades of Integration-Based, Geometric Vector Field Visualization,” *Proc. EuroGraphics’09*, pp. 73-92, (2009).

[2] A. Mebarki, P. Alliez, and O. Devillers, “Farthest Point Seeding for Efficient Placement of Streamlines,” *Proc. IEEE Visualization*, pp. 479–486, (2005).

[3] Z. Liu, R. J. MoorheadII, and J. Groner, “An Advanced Evenly-Spaced Streamline Placement Algorithm,” *IEEE Trans. Visualization and Computer Graphics*, vol. **12**, no. 5, pp. 965–972, (2006).

[4] W. Lefer, B. Jobard, and C. Leduc, “High-Quality Animation of 2D Steady Vector Felds,” *IEEE Trans. Visualization and Computer Graphics*, vol. **10**, no. 1, pp. 2–14, (2004).

[5] R. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. Post, and D. Weiskopf, “The State of The Art in Flow Visualization: Dense and Texture-Based Techniques,” *Computer Graphics Forum*, vol. **23**, no. 2, pp. 203–221, (2004).

[6] B. Cabral and L. C. Leedom, “Imaging Vector Fields Using Line Integral Convolution,” *Proc. ACM SIGGRAPH’93*, pp. 263–270, (1993).

[7] B. Jobard, G. Erlebacher, and M. Y. Hussaini, “Lagrangian-Eulerian Advection of Noise and Dye Textures for Unsteady Flow Visualization,” *IEEE Trans. Visualization and Computer Graphics*, vol. **8**, no. 3, pp. 211–222, (2002).

[8] S. Bachthaler and D. Weiskopf, “Animation of Orthogonal Texture Patterns for Vector Field Visualization,” *IEEE Trans. Visualization and Computer Graphics*, vol. **14**, no. 4, pp. 741–755, (2008).

[9] M.-T. Chi, T.-Y. Lee, Y. Qu, and T.-T. Wong, “Self-Animating Images: Illusory Motion Using Repeated Asymmetric Patterns,” *Proc. ACM SIGGRAPH’08*, pp. 1–8, (2008).

[10] W. T. Freeman, E. H. Adelson, and D. J. Heeger, “Motion Without Movement,” *Proc. ACM SIGGRAPH‘91*, pp. 27–30, (1991).

[11] A. Kitaoka, “Anomalous Motion Illusion and Stereopsis,” *Journal of Three Dimensional Images*, vol. **20**, pp. 9–14, (2006).

[12] G. Rong, T.-S. Tan, T.-T. Cao, and S. Indra, “Computing Two-Dimensional Delaunay Triangulation Using Graphics Hardware,” *Proc. Symposium on Interactive 2D Graphics and Games*, pp. 89-97, (2008).