Animating Streamlines with Orthogonal Advancing Waves

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

[*]Department of Computer Science and Information Engineering, National Cheng-Kung University, Taiwan, R.O.C.
[§]Department of Computer Science, Kentucky State University, Frankfort, KY 40601, USA
[ξ]NASA Ames Research Center, Moffett Field, CA 94035, USA

ABSTRACT

Self-animating image of flow through Repeated Asymmetric Patterns (RAPs) is an innovative approach for creating illusory motion using a single image. In this paper, we present a smooth cyclic variable-speed RAP animation model that emulates orthogonal advancing waves from a geometry-based flow representation. It enables dense accurate visualization of complex real world flows using animated streamlines of an elegant placement coupled with visually appealing orthogonal advancing waves. The animation model first performs velocity (magnitude) integral luminance transition on individual streamlines. Then, an inter-streamline synchronization in luminance varying along the tangential direction is imposed. Next, tangential flow streaks are constructed using evenly-spaced hue differing in the orthogonal direction. In addition, an energy-decreasing strategy is proposed that adopts an iterative yet efficient procedure for determining the luminance phase and hue of each streamline in HSL color space. To increase the contrast between flow streaks, an adaptive luminance interleaving in the direction perpendicular to the flow is further applied. We demonstrate the effectiveness of the animation model using some synthetic and real flows.

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

1. INTRODUCTION

An effective way to visualize numerical flow fields is to show particle traces. In steady flows, a streamline is a field line tangent to the velocity vector direction at an instant in time. The last two decades have seen significant research on flow visualization using streamlines [1, 2, 3, 4]. There has also been a great deal of work on other visualization techniques including geometry-/glyph-based methods [5] such as arrow plots and stream surfaces as well as texture-/image-based techniques [6], e.g., spot noise [7], Line Integral Convolution (LIC) [8], LEA [9], and IBFV [10]. While most of these techniques are usually used to provide a static visualization of the flow at an instantaneous time, the spatial-temporal features may not be sufficiently exposed. Animation is not only an instrumental tool for enhancing the display of spatial correlation such as flow direction, orientation (positive flow direction) [11], 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.

Flow animation using streamlines remains an effective way of revealing spatial-temporal structures not easily seen through a static visualization. However, there are several challenges for 2D flow animation. ***Spatial continuity***, favored in both tangential and orthogonal directions, facilitates mental reconstruction of the flow [12] even from a geometric representation [1, 2, 3, 4]. Insufficient continuity may prevent salient features from being recognized since an

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
David L. Kao: E-mail: David.L.Kao@nasa.gov
Tong-Yee Lee: E-mail: tonylee@mail.ncku.edu.tw
Color figures, images, and accompanying animations are available at http://graphics.csie.ncku.edu.tw/flowvis/

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 [12]. ***Temporal coherence***, a critical requirement imposed on texture synthesis [9, 10] and layout of geometric elements [3, 13], 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 avoided to produce an elegant cyclic animation. ***Variable-speed*** depiction offers further insight into the flow besides the directional information because velocity magnitude relates to such properties as momentum and density. ***The pattern of orthogonal waves*** [14] 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. [15] 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 challenges facing 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 [15] 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.

The remainder of this paper is organized as follows. Section 2 classifies existing animation techniques used in flow visualization and characterizes our RAPSA method in comparison with previous work. Section 3 presents the RAPSA method in detail, involving five components of the pipeline. Results are given in Section 4 to demonstrate the effectiveness of RAPSA in spatial continuity, high image contrast, temporal coherence, variable-speed delineation, and synthesis of orthogonal waves. We conclude the paper with a brief summary and outlook on future work.

2. RELATED WORK

Thoroughly reviewing previous work on vector field visualization and even specifically on flow animation is beyond the scope of this paper. We refer readers to two well-known sources [5, 6] for a comprehensive survey. Since flow animation techniques are sparsely scattered as part of flow visualization in the literature, we choose to discuss them here by means of classification with concise characterization and representative references, instead of long enumeration.

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 the size/length, position, and/or shape of geometric objects, e.g., massless particles [16] and deformed surfaces [17], to capture some snapshots of the physical 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] (though essentially the primitives themselves do not change or move at all). 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 modifying the phase of a periodic filter during image synthesis to create a sequence of frames for a cyclic animation [8, 18]. This kind of techniques exploits the local phase change of a stationary pattern to construct the sense of a global "motion without movement" [19]. 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 [9, 10, 14]. In general, one or more additional rendering passes are needed as a post-process to improve image contrast.

To our knowledge, Bachthaler and Weiskopf's approach [14] has been so far 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), *wherever possible*, 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 independent of yet amenable to GPU acceleration and possesses great potential of interactive flow visualization thanks to the recent advances in evenly-spaced streamline placement [1, 2]. Thus our method offers a novel 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 so good.

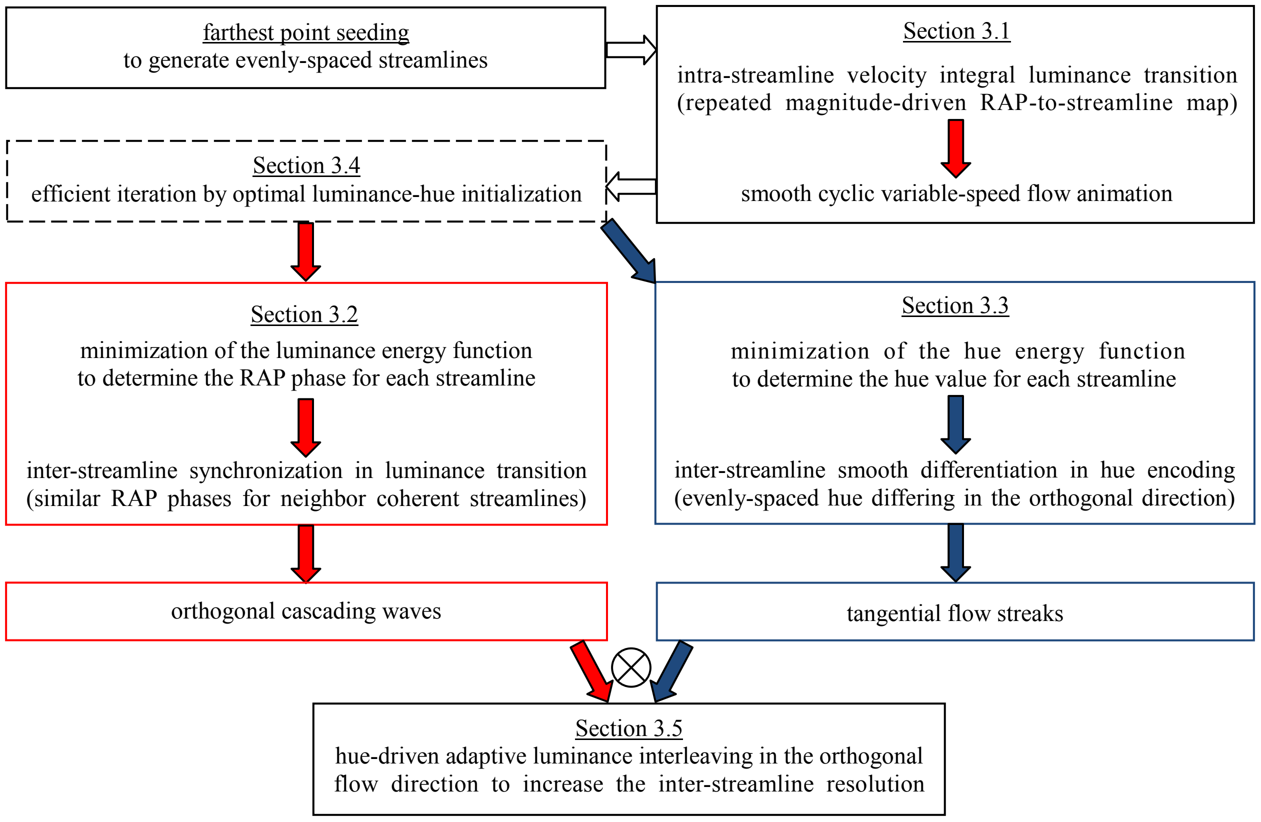
farthest point seeding
to generate evenly-spaced streamlines
Section 3.1
intra-streamline velocity integral luminance transition (repeated magnitude-driven RAP-to-streamline map)
smooth cyclic variable-speed flow animation
Section 3.4
efficient iteration by optimal luminance-hue initialization
Section 3.2
minimization of the luminance energy function to determine the RAP phase for each streamline
inter-streamline synchronization in luminance transition (similar RAP phases for neighbor coherent streamlines)
Section 3.3
minimization of the hue energy function to determine the hue value for each streamline
inter-streamline smooth differentiation in hue encoding (evenly-spaced hue differing in the orthogonal direction)
orthogonal cascading waves
tangential flow streaks
Section 3.5
hue-driven adaptive luminance interleaving in the orthogonal flow direction to increase the inter-streamline resolution

Figure 1. The pipeline of RAPSA where Section 3.4 essentially serves as the implementation of Sections 3.2 and 3.3.

3. RAP-BASED STREAMLINES ANIMATION

This section describes our RAP-based streamlines animation algorithm, i.e., RAPSA, which employs the farthest point seeding method [1] to place evenly-spaced long streamlines. Specifically, the seed of any new streamline is positioned to be the farthest away from all existing streamlines. By rendering the Delaunay triangles of the streamline layout, RAPSA emulates a dense representation of the flow. Thus the essential task is to determine the specific color for every point of the evenly-spaced streamlines. 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 primary RAP animation model that uses velocity integral luminance transition to achieve smooth cyclic variable-speed flow motion (Section 3.1). What follows is the extension of this model by synchronizing neighboring streamlines, *wherever possible* (contingent on the local coherence involving velocity magnitude and flow direction such as convergence/divergence etc), in luminance varying along the tangential flow direction to form orthogonal propagating waves (Section 3.2). Then we refine the model further with evenly-spaced hue differing in the orthogonal direction for exposing tangential flow streaks (Section 3.3). To combine these two mutually-perpendicular families of patterns, an energy-decreasing strategy is presented to obtain the specific luminance phase and hue of each streamline by rapid iteration (Section 3.4). We conclude the pipeline with adaptive luminance interleaving to augment the inter-streamline contrast initially established by hue differing (Section 3.5). Figure 1 shows the components of the RAPSA pipeline. Table 1 lists the major functions that we define and use to describe the pipeline in detail.

Functions	Definitions	Sections
$L_s(n)$	the luminance of point *n* on streamline *s*	3.1
$L_{st}(n)$	the luminance of point *n* on streamline *s* at time step *t* (or in frame *t*)	3.1
$\ell_s(n)$	the velocity (or magnitude-driven) integral luminance transition or the accumulated luminance change from the starting point to point *n* along streamline *s*	3.1
$\Phi(L_s(n))$ $\Psi(L_s(n))$	the luminance values of the two orthogonally immediate neighbors of streamline *s* at point *n*, respectively	3.2
$\Delta(\Phi, L, s, n)$ $\Delta(L, \Psi, s, n)$	the differences in luminance between streamline *s* and its two orthogonally immediate neighbors at point *n*	3.2
$E(L)$	an energy term for measuring the overall inter-streamline luminance incoherence	3.2
$H_s(n)$	the hue of point *n* on streamline *s*	3.3
$\Phi(H_s(n))$ $\Psi(H_s(n))$	the hue values of the two orthogonally immediate neighbors of streamline *s* at point *n*, respectively	3.3
$\Delta(\Phi, H, s, n)$ $\Delta(H, \Psi, s, n)$	The differences in hue between streamline *s* and its two orthogonally immediate neighbors at point *n*	3.3
$E(H)$	an energy term for measuring the overall inter-streamline hue incoherence	3.3

Table 1. Major functions used in the pipeline description.

3.1 Creating Cyclic Variable-Speed Animation

Intensity tapering is an effective motion metaphor and has been used in oriented LIC [11]. We adopt TYPE I RAP [20] as the intensity ramp that is repeatedly placed along streamlines to indicate the flow orientation. The motion direction conveyed by TYPE I RAP refers to the dark-to-bright transition. To make efficient use of the color spectrum, we choose HSL (hue, saturation, and lightness or brightness) color space for mapping the RAP brightness (or *luminance* hereinafter) 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 in a normalized range [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 raw, monotonically increasing luminance to [0, 1), i.e., the range of the actual, valid luminance that bounds every full RAP. With this luminance range, a smaller value of f results in a (spatially) longer RAP and vice versa, while a streamline carries or is attached with multiple sequentially connected RAPs (Fig. 2). 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 (or inter-frame shift) and hence a higher speed (that the viewer perceives) 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 any streamline present in the scene, for 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 $1/T$ per frame. 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 $t \in [0, T\text{-}1]$ is an integer indicating the frame index and $L_{st}(n)$ is the spatial-temporal description of the luminance. By fixing parameter f for all points throughout the flow domain, the same RAP is then repeatedly placed along all streamlines, producing a constant-speed animation. By adjusting f based on individual points with the smallest value for the highest speed and the largest for the lowest, many RAPs of different lengths are mapped to the streamlines, yielding a variable-speed animation. Fig. 2 demonstrates the use of parameter f for constant-speed (Fig. 2a) and variable-speed (Fig. 2b) animation RAPs.

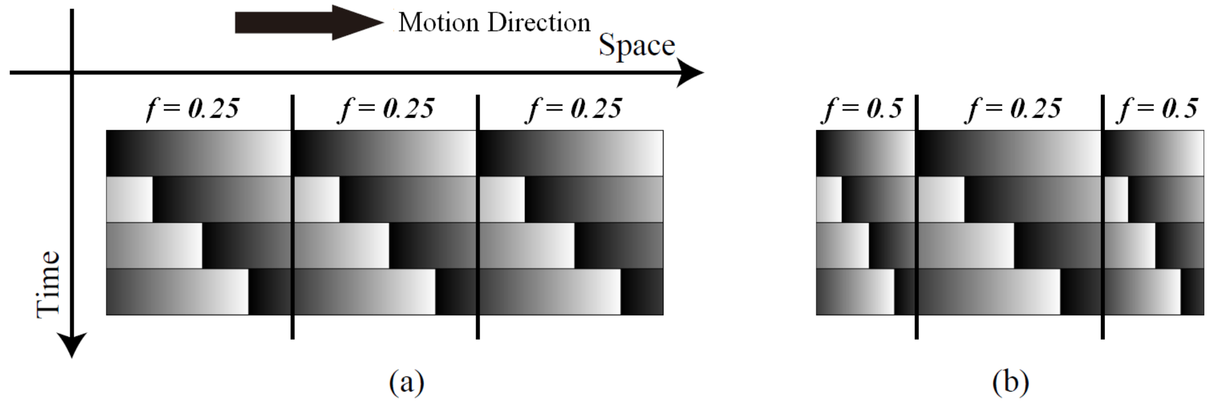

Figure 2. RAPs and two illustrative settings of parameter f for generating (a) constant-speed and (b) variable-speed motion effects, respectively. With the full span of each RAP accessed uniformly by four successive frames (indexed downward), the longer RAP by the smaller value of f (0.25) exhibits a greater inter-frame spatial change and hence a higher speed perceived in the resulting animation than the shorter RAP by the larger value of f (0.50) does. Note that the motion direction conveyed by each RAP points to the right as the luminance transitions from the lowest (black) to the highest (white) to create Fraser-Wilcox illusion [20].

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 for generating a 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 calculated 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 $\beta_{max,}$ $0<\beta_{min}<\beta_{max}<\pi/2$, are two threshold angles used in a monotonically decreasing function $\delta_s(n)$ to inversely map the lowest velocity magnitude to $\tan(\beta_{max})$ and the highest to $\tan(\beta_{min})$. $\delta_s(n)$, a point-wise 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 physically reasonable 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 [14] 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 separately applying RAP maps to individual streamlines (Fig. 3a). According to Equation (3), it suffices to establish spatial correlation between streamlines in $L_s(n)$. Specifically 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 resulting synchronization depends on the local coherence between neighboring streamlines in both flow direction (e.g., convergence/divergence etc) and velocity magnitude.

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

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

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

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

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 incoherence. 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. 3b).

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. 3b). 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 across 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. 3c) 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

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

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

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

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 incoherence. 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.

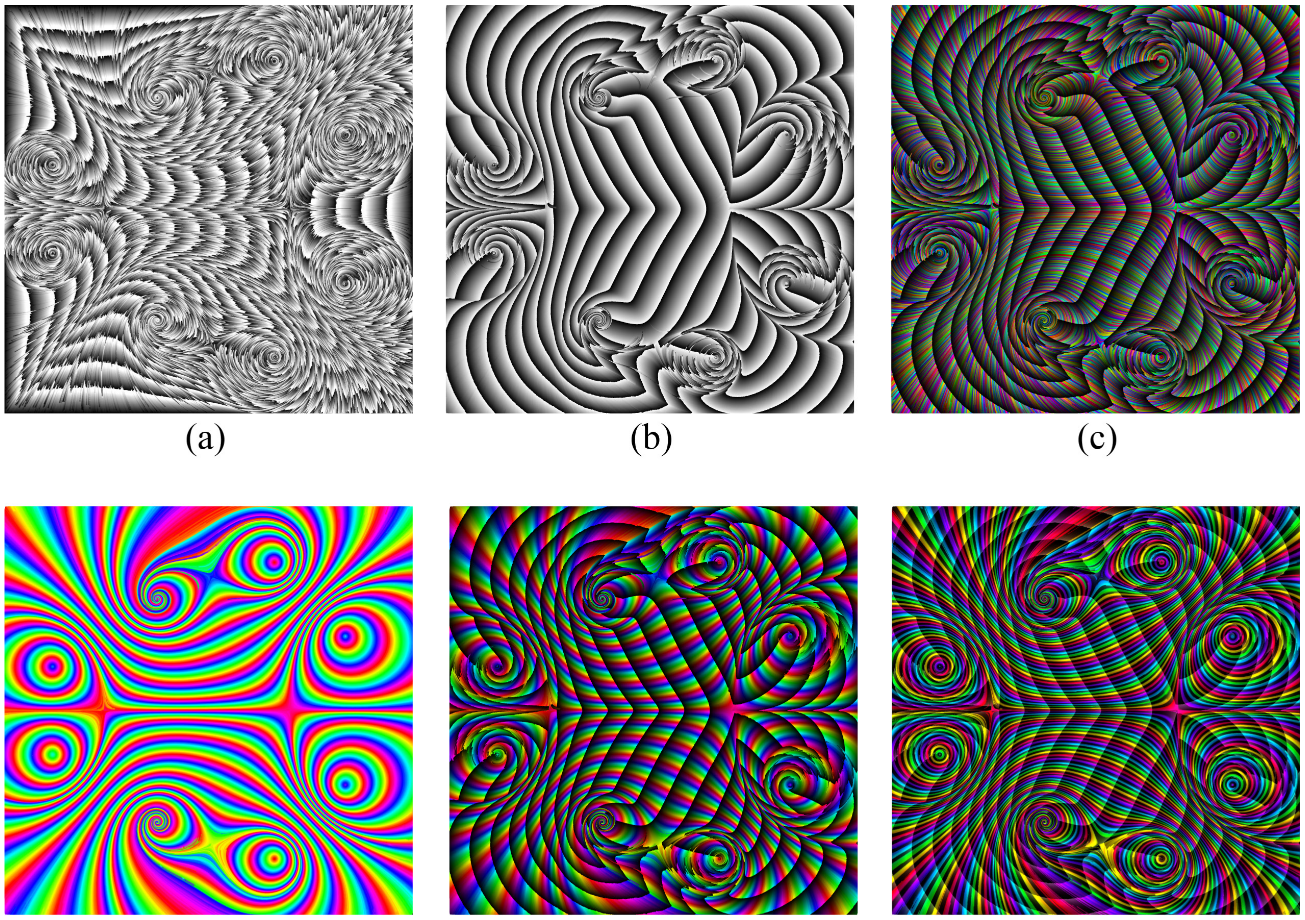

Figure 3. Synthesis of orthogonal cascading waves and construction of tangential flow streaks. (a) As streamlines are separately mapped with the RAP luminance, RAP segments overlap one another to form an unorganized pattern. This layout not only fails to build clear orthogonal strips but also affects 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.

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

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

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

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 smoothes 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 the longest (orthogonal) curve that is perpendicular to some evenly-spaced streamlines, of which the relatively middle 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 (these streamlines are numbered along the orthogonal curve). 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. 4.

Fig. 3d demonstrates the sole use of smooth evenly-spaced hue differing for constructing tangential flow patterns. Fig. 3e 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 must be small enough to suppress distracting artifacts, the visual granularity tends to lead to insufficient image contrast (Fig. 3d and Fig. 3e). 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 an artificial 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

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

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

```
TS and OS:   set of evenly-spaced Tangential Streamlines and set of evenly-spaced Orthogonal Streamlines, respectively.
STS:         queue of Source Tangential Streamlines ∈ TS,   each for determining an orthogonal value-adjustment path.
h:           a user-specified inter-streamline hue interval.
line status: UNINITED (θ and H not yet inited) / INITED (possibly to be further updated) / DONE (no further update).

foreach ( streamline ts in TS ) ts.status = UNINITED;

foreach ( streamline ts in TS )
{   if ( ts.status != UNINITED ) continue;                    // only an uninitialized streamline can serve as a source
    ts.θ = 0;          ts.H = 0;          ts.status = INITED;          STS.push( ts );

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

        set of streamlines tsLines = sts plus the tangential streamlines ∈ TS   that intersect with os;
        streamlines in tsLines are spatially numerated or indexed (along os) from 0 to (nLines – 1);

        for ( i = 0; i < nLines; i ++ )
        {   sample refers to the intersection between os and tsLines[i] and it lies between points m and n along tsLines[i];
            lumin[i]  = interpolation between  $\ell_{tsLines[i]}(m)$ and  $\ell_{tsLines[i]}(n)$ for sample;   // Equation 5 for $\ell_s(n)$
            LUMIN[i] = interpolation between $L_{tsLines[i]}(m)$ and  $L_{tsLines[i]}(n)$ for sample;   // Equation 6 for $L_s(n)$
        }

        // take the roughly-middle streamline as a reference and   adjust θ and H for other lines in a propagating manner
        referenceId = Id of an initialized tangential streamline that is the closest to index nLines/2;
        tsLines[referenceId].status = DONE;

        for ( i = referenceId; i > 0; i -- )                    // update θ and H for the streamlines to “left” of the reference
        if ( tsLines[i-1].status != DONE )
        {   tsLines[i-1]. θ = mod( LUMIN[i] – lumin[i-1] );        tsLines[i-1].H = mod( tsLines[i].H - h );
            tsLines[i-1].status = DONE;                           if ( tsLines[i-1] != ts ) STS.push( tsLines[i-1] );
        }

        for ( i = referenceId; i < nLines – 1; i ++ )           // update θ and H for the streamlines to “right” of the reference
        if ( tsLines[i+1].status != DONE )
        {   tsLines[i+1]. θ = mod( LUMIN[i] – lumin[i+1] );        tsLines[i+1].H = mod( tsLines[i].H + h );
            tsLines[i+1].status = DONE;                           if ( tsLines[i+1] != ts ) STS.push( tsLines[i+1] );
        }
    }
}
```

Figure 4. Pseudo code for initializing the phase and hue values of streamlines.

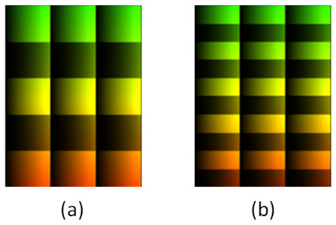

Figure 5. The role of parameter γ in determining the resolution of luminance interleaving on a pattern created by the combination of synchronized luminance transition and evenly-spaced hue differing (with a certain hue interval h) from a horizontal flow. (a) Five zebra ribbons come into view when γ is 10. (b) Ten zebra ribbons are discernible when γ is 20. The bright ribbons result from the hue-low-pass band and the dim from the luminance suppression band.

Fig. 5 illustrates the effect of parameter γ on the density of luminance interleaving for 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 (γ) in the luminance component and the other (h) in the hue component. Their multiplication controls the ultimate inter-streamline differentiation. Fig. 6 portrays the correlation between these two parameters in contributing to the inter-streamline contrast. Satisfactory results are obtained by using h = 0.01 and γ = 20 in this paper. Equation (15) may be modified regarding the threshold (0.5) and the luminance suppression function to produce specific visual effects.

Fig. 3f 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. 3f than in Fig. 3e, increasing the overall image contrast while avoiding artifacts.

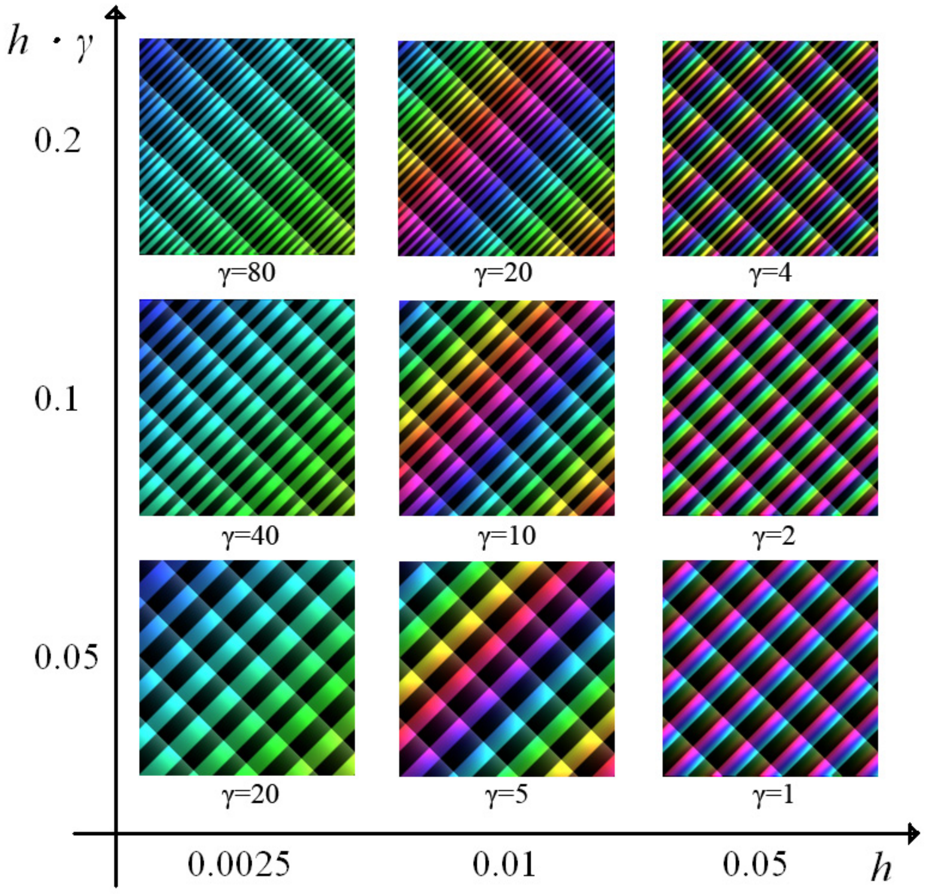

Figure 6. The correlation between the hue interval h (Section 3.4) and the luminance interleaving parameter γ (Section 3.5) in the overall inter-streamline contrast of the visualization of a diagonal flow. The images along each column (with h fixed) reflect the influence of γ and the images along each row indicate that of various combinations of h and γ (with their multiplication fixed) on differentiating streamlines.

4. RESULTS AND DISCUSSIONS

We have implemented our RAPSA algorithm using Visual C++ on a PC (Intel i7 2.8GHz and 4GB RAM) running Windows Vista. This section gives some results and discussions about the computational performance and animation quality. Further provided is a perceptual analysis of RAPSA as it is compared to two LIC variants.

4.1 Computational Performance

With 1.5% (relative to the width of a flow field) as the streamline density for the results reported in section 4, the overall cost for producing an animation of several frames involves tangential streamline placement [1], orthogonal streamline placement, Delaunay triangulation, RAPSA core modules (determining the luminance phase and hue for every streamline), phase shifting, triangles rendering, and post-processing. The former four stages are executed only once for all frames of an animation, whereas the latter three stages are performed for per-frame generation. Table 2 gives the time breakdowns (in seconds) for creating several 8-frame RAPSA animations, of which each has one frame shown in Fig. 8 ~ Fig. 10, respectively.

It is worth mentioning that the computational performance varies with the streamline density and the data size. Specifically, the computational cost of evenly-spaced streamline placement (for both tangential and orthogonal streamlines) increases as the density does [1, 2, 3, 13]. Given a density, the total number of (streamline) points is proportional to the data size, while the cost of Delaunay triangulation, i.e., one major bottleneck of the RAPSA pipeline, increases with the number of points. It is also the case with the time used for per-frame generation in relation to the number of points. Thus the time breakdowns given in Table 2 result from a high streamline density and medium-to-large flow fields. The computational performance will be higher when lowering these requirements and testing RAPSA on a decent platform. Currently only the Delaunay triangulation stage is implemented using CUDA [21] on GPU. Thus there is a great potential for further accelerating RAPSA, e.g., by implementing the other stages with CUDA and placing evenly-spaced tangential and orthogonal streamlines with a faster algorithm such as ADVESS [2]. Since ADVESS exports non-uniform points along each streamline due to the use of an adaptive step size, it needs to be extended for providing evenly-spaced samples along each streamline as the input of RAPSA.

4.2 Animation Quality

As detailed in Section 1, there are several important factors 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 of this section (http://graphics.csie.ncku.edu.tw/flowvis/).

Figure and Data Size	Stages Executed Once for All Frames				Per-Frame Generation	Total Time (8 frames)
	Tangential Streamline Placement	Orthogonal Streamline Placement	Delaunay Triangulation of Streamlines	RAPSA Core Modules	Phase-Shifting Triangles Rendering Post-Processing	
8a (400 × 400)	0.80	0.47	0.98	0.53	0.52	6.94
8b (400 × 400)	0.83	0.53	0.98	0.55	0.50	6.89
8c (400 × 400)	0.76	0.45	0.95	0.53	0.50	6.69
8d (400 × 400)	0.75	0.52	0.98	0.56	0.50	6.81
8e (400 × 400)	0.76	0.52	1.01	0.56	0.51	6.93
8f (400 × 400)	0.84	0.52	0.98	0.55	0.50	6.89
9b (576 × 291)	0.84	0.36	0.97	0.53	0.56	7.18
10b (600 × 400)	1.53	0.48	0.70	0.47	0.38	6.22

* Test platform: Intel i7 2.8GHz / 4GB RAM / Windows Vista.

* Streamline density: 1.5%.

* Tangential streamline placement, orthogonal streamline placement, Delaunay triangulation of streamlines, and RAPSA core modules are executed once for all frames to result in a base cost while the per-frame generation cost times the number of frames (8) adds to the total time.

Table 2. Time breakdowns (in seconds) for creating several 8-frame RAPSA animations.

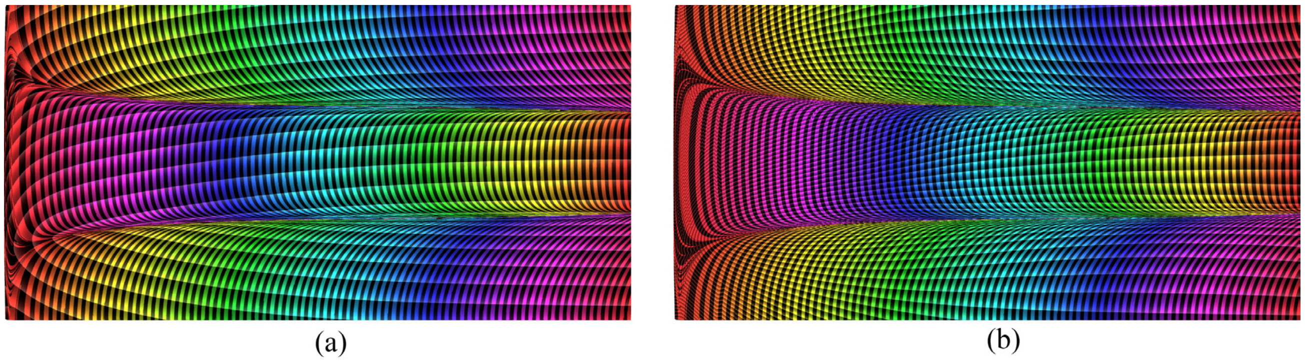

Figure 7. Two RAPSA images generated in (a) constant-speed mode and (b) variable-speed mode, respectively, for visualizing a 770 × 386 flow field. Smooth evenly-spaced hue differing, combined with adaptive luminance interleaving, accentuates individual flow streaks in HSL space to yield high inter-streamline 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, constant in (a) but variable in (b), encodes the velocity magnitude.

Fig. 7 shows two RAPSA images produced in constant-speed mode (the simplified mode, Fig. 7a) and variable-speed mode (the standard mode, Fig. 7b), 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. 7a, constant-size RAPs are employed for streamlines, resulting in a constant-speed flow animation. Fig. 7b 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 orthogonal curves 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. 8 shows six RAPSA images generated by visualizing 400 × 400 synthetic 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 the constituent critical points. Evenly-spaced hue differing offers great aid in the analysis of the underlying interaction between the 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 topologically-coherent regions. In fact, the magnitude (in terms of the length and width) of orthogonal waves is intrinsically influenced by and hence directly reflects the degree of local flow convergence and divergence (Section 3.2).

Fig. 9 shows two images generated using SAI [15] and RAPSA, respectively, for a 576 × 291 wind flow field. The SAI image (Fig. 9a) conveys the flow in a very discrete sparse 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. 9b) gives a dense flow representration. The curve segments, aligned along the cascading strips, can be successively connected one by one in the tangential direction 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.

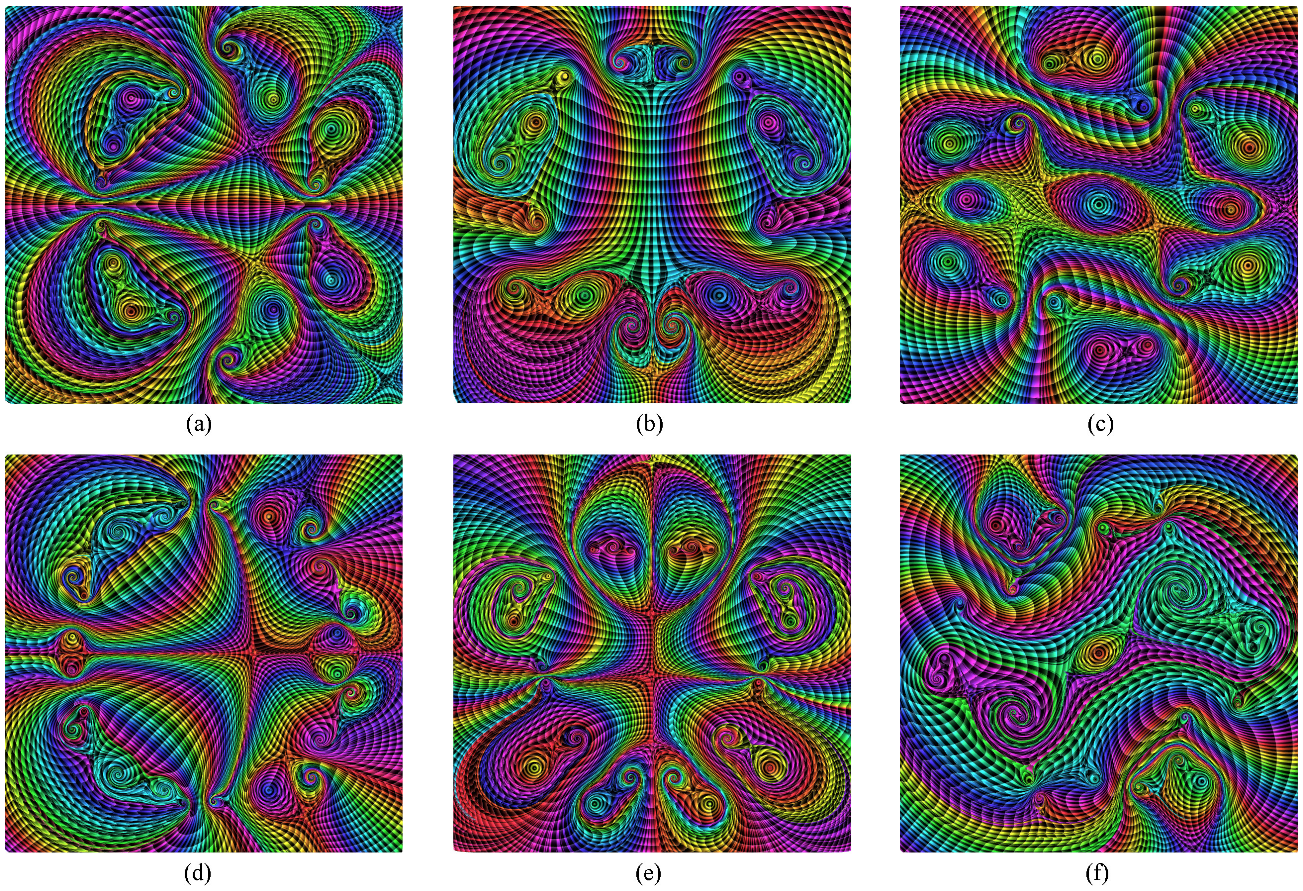

Figure 8. RAPSA images for the visualization of two x-axis symmetric (left column), two y-axis symmetric (middle column), and two center-based symmetric (right column) 400 × 400 complex flow datasets.

Fig. 10 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. 10a) and standard (Fig. 10b) evenly-spaced hue differing. In the former case, normal hue differing (Section 3.3) 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. 10a is still able to exhibit flow streaks and the blueness mimics the real world ocean apperance. Fig. 10b 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. 11 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 compose the images and the underlying land and sea. Given a set of opacities increasing from 0 (Fig. 11a) to a fraction (Fig. 11b) and further untill 1 (Fig. 11c) and then decreasing back to 0 (Fig. 11a), the entire multi-cycle animation associates the flow pattern (e.g., the eye of Isabel at the center) with the physical location. Though created for a single time step of the data, this cyclic animation can provide important clues for investigating the trend of hurricane evolution.

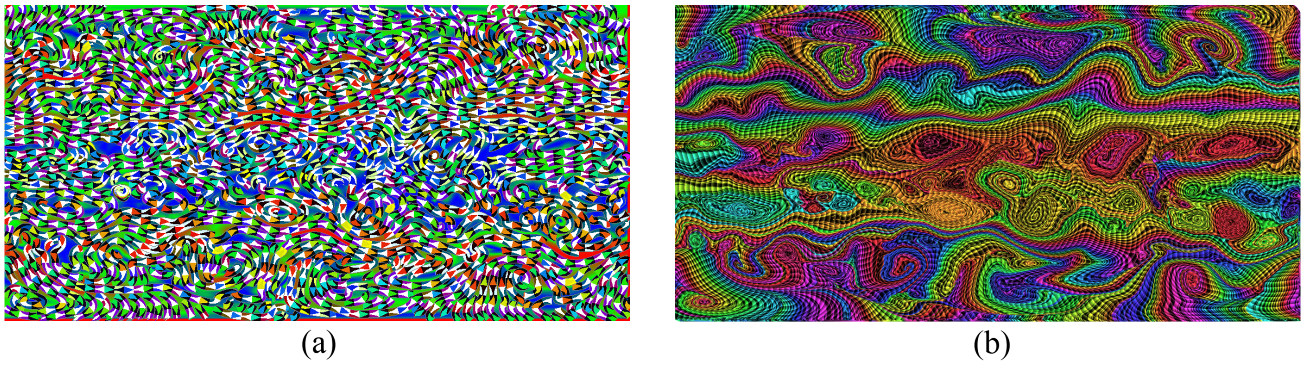
Figure 9. Comparison of SAI [15] and RAPSA in displaying a 576 × 291 wind flow. The SAI image (a) suffers from low spatial resolution and glyph cluttering. The RAPSA image (b) provides a dense representation of the flow in both the tangential/integral direction and the orthogonal direction, while flow streaks are still easily discernible.

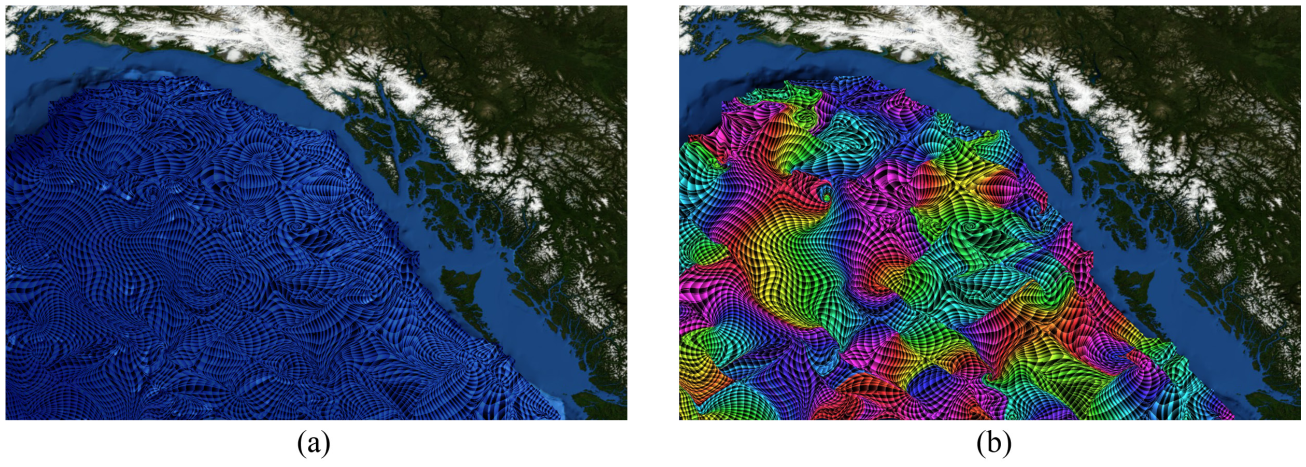
Figure 10. Two RAPSA images for visualizing the Northeast Pacific Ocean flow. Both followed by luminance interleaving, (a) 'degenerate' evenly-spaced hue differing imitates the real ocean flow with blueness as the single hue and (b) standard hue differing is intended to emphasize the insight.

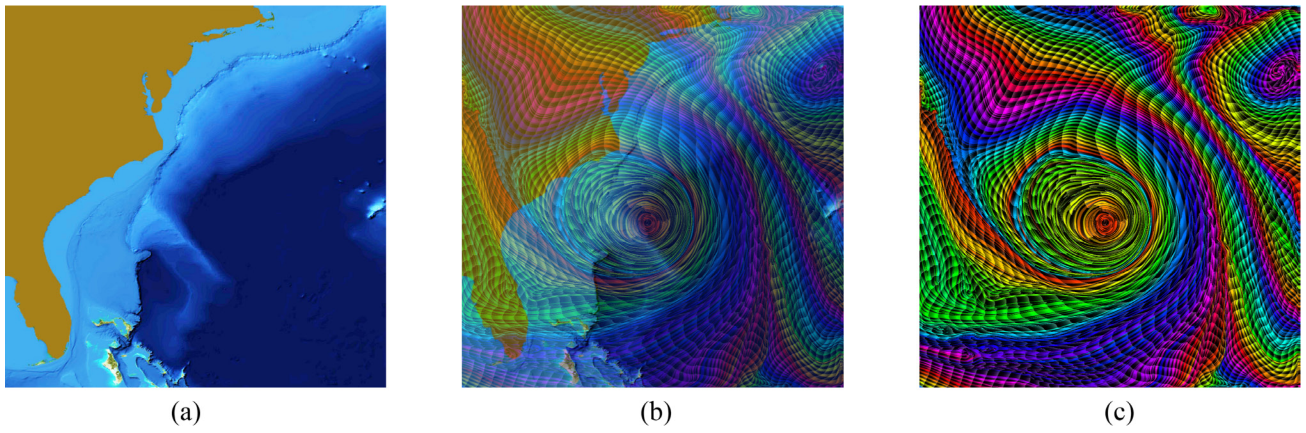
Figure 11. Three snapshots of a multi-cycle animation of RAPSA images, blended with the context through a series of opacities that increase from 0 (a) to a fraction (b) and further until 1 (c) and then decrease back to 0 (a), for visualizing a 2D slice of Hurricane Isabel at time step 24. An individual image captures the instantaneous structure of Isabel and the animation helps explore its development.

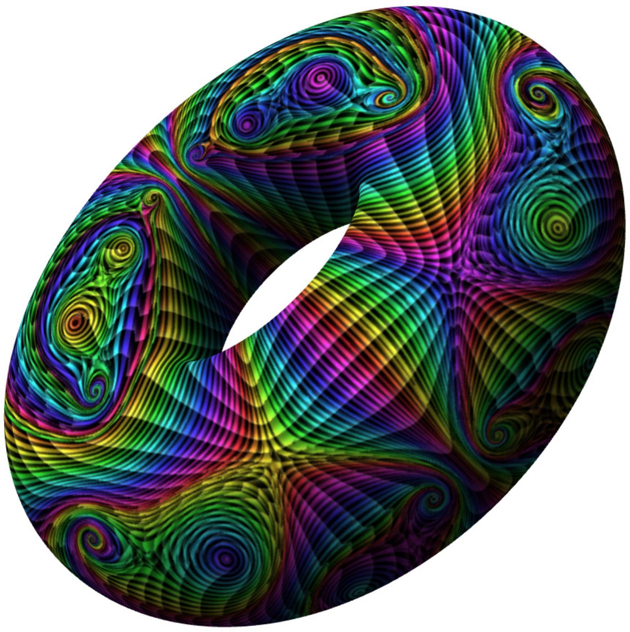

Figure 12. The application of RAPSA to the visualization of a surface flow synthesized on a 400 × 400 curvilinear grid of torus. The orthogonal advancing waves help enhance direction cueing as well as depth cueing for a better understanding of the flow movement in 3D settings.

Fig. 12 illustrates the use of RAPSA for visualizing the motion of a complex surface flow synthesized on a 400 × 400 curvilinear grid of torus. Eight RAPSA frames are generated in 2D image space before being texture-mapped onto the surface with a tilted view. The orthogonal waves, propagating along the tangential flow direction, not only distinguish between laminar and turbulent areas for highlighting the salient flow features but also help understand the shape of the surface.

4.3 Perceptual Analysis

Fig. 13 compares RAPSA against oriented LIC [11] and enhanced LIC [22] in visualizing a flow with three circular structures. Like oriented LIC, RAPSA exploits intensity tapering to exhibit flow orientation, showing an advantage over enhanced LIC. As oriented LIC employs a single ramp to depict each of a set of randomly “selected” short segments of the flow, RAPSA is able to delineate a very long streamline by using multiple ramps (i.e., RAPs) along the curve. Thus RAPSA is more effective than oriented LIC in describing the tangential flow direction because of the correlation between the RAPs in sequence. However, inter-RAP luminance discontinuities more or less hinder the visual reconstruction of a streamline, particularly near turbulent flow areas (e.g., around critical points). The ramp space requirement implies that RAPSA is not well suited for displaying short streamlines which may also be placed near turbulent flow areas. The essence of evenly-spaced streamlines is a discrete sparse representation in the direction perpendicular to the flow. On one hand, this seems to provide a lower orthogonal resolution than that achieved by enhanced LIC and the dim ribbons resulting from adaptive luminance interleaving may possibly be a distraction. On the other hand, a sufficiently high streamline density guarantees a strong coherence between neighboring streamlines, justifying and facilitating visual interpolation across the flow from an aesthetic layout to approach a dense representation. It is worth mentioning that flow streaks are actually embedded in an enhanced LIC image and therefore only short jaggy curve segments can be assimilated from correlated pixels — a synthesized approximation to the field lines, whereas flow streaks conveyed by RAPs in an RAPSA image are a smooth *accurate* clear-cut representation of the field lines. In addition, zebra ribbons indeed contribute to high image contrast for RAPSA.

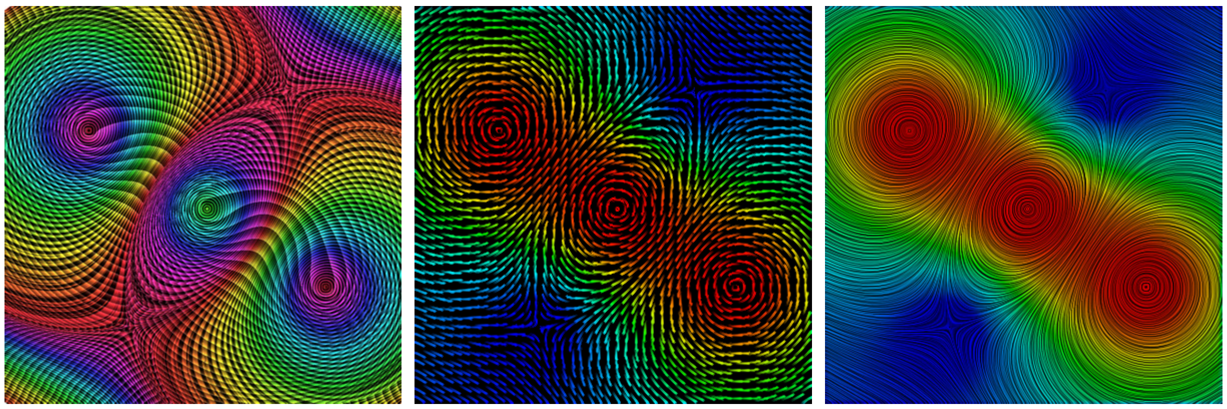
(a) (b) (c)

Figure 13. Comparison of (a) RAPSA with (b) oriented LIC and (c) enhanced LIC in visualizing a flow with three circular structures. A rainbow color map is applied to (b) and (c), with blue for the lowest velocity magnitude and red for the highest. Enhanced LIC cannot reveal the flow orientation in a stationary image. RAPSA outperforms the two LIC variants in showing speed variety both spatially (in images) and temporally (in animations) — without the help of color maps, representing long streamlines accurately, and using an auxiliary visualization metaphor — orthogonal waves.

Besides the ability to show flow orientation and speed variety in a stationary image, RAPSA is advantageous in displaying orthogonal waves. One side effect (Fig. 14) is that such auxiliary curves might cause confusion and distraction to the understanding of tangential directions, and even worse, mislead the viewer at first glance into assuming them as tangential flow streaks. This is particularly the case with flow visualization users who are new to orthogonal waves. Nevertheless, those familiar with propagating patterns tend to realize that they, as widely seen from ocean flows, offer a supplemental perspective to improve the perception of tangential directions. In fact, RAPSA is primarily dedicated to generating flow animations rather than showing separate stationary images. In other words, the merits may be well appreciated in animations where any side effect may be significantly mitigated. This incurs a necessity for a rigorous user study [12] in our future work to evaluate RAPSA on human perception.

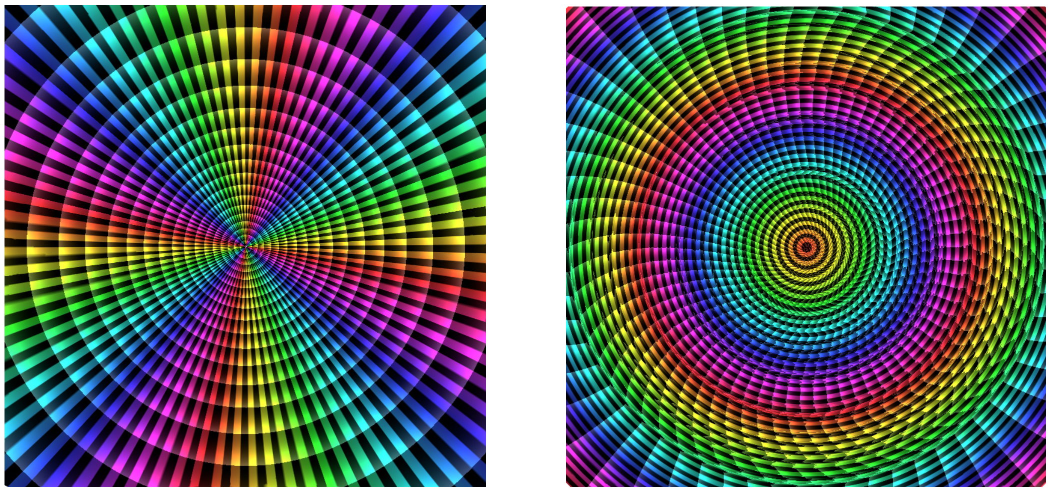
(a) (b)

Figure 14. One possible side effect that orthogonal waves may cause by confusing and distracting the viewer as tangential directions are investigated for two synthetic flows with (a) a source structure and (b) a clockwise circular structure, respectively, using stationary RAPSA images. This side effect turns out to be low to none in RAPSA animations.

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 work [15], enabling dense accurate visualization of complex real world flows using animated streamlines of an elegant placement coupled with visually appealing orthogonal advancing waves.

As for future work, we may accelerate the RAPSA algorithm regarding various components of the pipeline. More importantly, we would like to extend our approach for time-varying flows and enhance it for complex curved surface flows (e.g., defined on an arbitrary triangular mesh) using an image-space oriented strategy. The former requires some changes to streamline placement to address temporal coherence [3] and the latter involves projecting surface flow vectors to 2D image space as well as some view-dependency and geometry edge issues [23]. We are also interested in their combination, i.e., visualization of unsteady surface flows.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported in part by 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] A. Mebarki, P. Alliez, and O. Devillers, "Farthest Point Seeding for Efficient Placement of Streamlines," *Proc. IEEE Visualization'05*, pp. 479-486, 2005.

[2] 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.

[3] B. Jobard and W. Lefer, "Unsteady Flow Visualization by Animating Evenly Spaced Streamlines," *Proc. EuroGraphcs'00*, pp. 21-31, 2000.

[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] 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.

[6] 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.

[7] J. J. van Wijk, "Spot Noise: Texture Synthesis for Data Visualization," *Proc. ACM SIGGRAPH'91,* pp. 309-318, 1991.

[8] B. Cabral and L. C. Leedom, "Imaging Vector Fields Using Line Integral Convolution," *Proc. ACM SIGGRAPH'93*, pp. 263-270, 1993.

[9] 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.

[10] J. J. van Wijk, "Image Based Flow Visualization," *Proc. ACM SIGGRAPH'02,* pp. 745-754, 2002.

[11] R. Wegenkittl, E. Groller, and W. Purgathofer, "Animating Flow Fields: Rendering of Oriented Line Integral Convolution," *Proc. Computer Animation'97*, pp. 15-21, 1997.

[12] Z. Liu, S. Cai, J. E. Swan II, R. J. Moorhead II, J. P. Martin, and T. J. Jankun-Kelly, “A 2D Flow Visualization User Study Using Explicit Flow Synthesis and Implicit Task Design,” *IEEE Trans. Visualization and Computer Graphics*, vol. 18, no. 5, pp. 783-796, 2012.

[13] Z. Liu and R. J. Moorhead II, “Interactive View-Driven Evenly Spaced Streamline Placement,” *Proc. IS&T / SPIE Conference on Visualization and Data Analysis (VDA '08)*, 68090A, pp. 1-12, 2008.

[14] 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.

[15] 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.

[16] A. J. S. Hin and F. H. Post, “Visualization of Turbulent Flow with Particles,” *Proc. IEEE Visualization '93*, pp. 46-53, 1993.

[17] C. Silva, L. Hong, and A. Kaufman, “Flow Surface Probes for Vector Field Visualization,” *Scientific Visualization — Overviews, Methodologies, and Techniques (Editors G. M. Nielson, H. Hagen, and H. Muller)*, IEEE Computer Society, pp. 295-310, 1997.

[18] D. Stalling and H.-C. Hege, “Fast and Resolution Independent Line Integral Convolution,” *Proc. ACM SIGGRAPH '*95, pp. 249-256, 1995.

[19] W. T. Freeman, E. H. Adelson, and D. J. Heeger, “Motion Without Movement,” *Proc. ACM SIGGRAPH '91*, pp. 27-30, 1991.

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

[21] 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.

[22] A. Okada and D. L. Kao, “Enhanced Line Integral Convolution with Flow Feature Detection,” *Proc. IS&T/SPIE Electronics Imaging '97*, pp. 206-217, 1997.

[23] B. Spencer, R. S. Laramee, G. Chen, and E. Zhang, “Evenly-Spaced Streamlines for Surfaces: An Image-Based Approach,” *Computer Graphics Forum*, vol. 28, no. 6, pp. 1618-131, 2009.