A 2D Flow Visualization User Study Using Explicit Flow Synthesis and Implicit Task Design

Zhanping Liu, Shangshu Cai, *Member, IEEE*, J. Edward Swan II, *Member*, *IEEE*,
Robert J. Moorhead II, *Senior Member, IEEE*, Joel P. Martin, T. J. Jankun-Kelly, *Member, IEEE*

Abstract—This paper presents a 2D flow visualization user study that we conducted using new methodologies to increase the objectiveness. We evaluated grid-based variable-size arrows, evenly spaced streamlines, and LIC variants (basic, oriented, and enhanced versions) coupled with a colorwheel and/or rainbow color map, which are representative of many geometry-based and texture-based techniques. To reduce data-related bias, template-based explicit flow synthesis was used to create a wide variety of symmetric flows with similar topological complexity. To suppress task-related bias, pattern-based implicit task design was employed, addressing critical point recognition, critical point classification, and symmetric pattern categorization. In addition, variable-duration and fixed-duration measurement schemes were utilized for lightweight precision-critical and heavyweight judgment-intensive flow analysis tasks, respectively, to record visualization effectiveness. We eliminated outliers and used the Ryan REGWQ post-hoc homogeneous subset tests in statistical analysis to obtain reliable findings. Our study shows that a texture-based dense representation with accentuated flow streaks, such as enhanced LIC, enables intuitive perception of the flow, while a geometry-based integral representation with uniform density control, such as evenly spaced streamlines, may exploit visual interpolation to facilitate mental reconstruction of the flow. It is also shown that inappropriate color mapping (e.g., colorwheel) may add distractions to a flow representation.

Index Terms—Flow visualization, user study, visualization effectiveness, flow synthesis, task design, test strategy, LIC, evenly spaced streamlines.

1 Introduction

Flow visualization seeks to provide insight into flow patterns for visual data analysis and plays a crucial role in oceanographic-atmospheric modeling, computational fluid dynamics (CFD) simulation, and electromagnetic field analysis. Texture-/image-based visualization methods [1] such as Line Integral Convolution (LIC) [2] are gaining considerable attention due to the dense continuous representation, compared to the sparse discrete representation employed by geometry-/glyph-based techniques [3] like arrows and streamlines. Different visualization techniques may be advantageous in different flow analysis tasks. Of the many flow visualization algorithms, only a few have been evaluated to determine their effectiveness [4], [5]. As a consequence, the best methods may not have been incorporated into visualization systems. This situation prevents domain scientists from unleashing the power of emerging techniques to explore complex flow phenomena. Without their feedback, visualization researchers may not realize practical needs to improve existing algorithms or find new problems to devise innovative methods. Thus more user studies are needed to better understand the relative merits of each flow visualization technique [6].

As stated in the 2006 NIH-NSF report on Visualization Research Challenges [7], user studies are important in the research, development, and deployment of flow visualization. A lot of work needs to be done to resolve bias issues that may occur through various stages or components of a user study. Without bias avoidance methodologies, a flow visualization user study could be heavily compromised. There is more to a flow visualization user study than the scenarios (e.g., surface, volume, and time-varying flows) being considered, the techniques (e.g., IBFV [8], IBFVS [9], and ISA [10]) being evaluated, the flow features (e.g., separation, attachment, and vortex core) being examined, and the specific yet usually ad-hoc findings being obtained. Thus conducting objective 2D flow visualization user studies, even with traditional and well-known techniques, remains an open problem. The valid methodologies gained from such efforts will not only refine our understanding of some (well-known) 2D flow visualization techniques, e.g., by offering quantitative support for qualitative evidence or anecdotal advice, but also and more importantly help formulate a general framework that is necessary for carrying out convincing flow visualization user studies with more complex configurations.

Motivated by the necessity for and significance of ef-

- *Zhanping Liu is with the Department of Radiology, University of Pennsylvania, 3600 Market Street, Suite 380, Philadelphia, PA 19104. E-Mail: zhanpingliu@hotmail.com.*
- *Shangshu Cai is with the Center for Risk Studies and Safety, University of California at Santa Barbara, 6740 Cortona Dr., Goleta, CA 93117. E-Mail: raymondcai@engineering.ucsb.edu.*
- *J. Edward Swan II and T. J. Jankun-Kelly are with the Department of Computer Science and Engineering, Mississippi State University, Mississippi State, MS 39762. E-Mail: {swan, tjk}@acm.org.*
- *Robert J. Moorhead II is with the Geosystems Research Institute and the Electrical and Computer Engineering Department, Mississippi State University, Mississippi State, MS 39762. E-Mail: rjm@gri.msstate.edu.*
- *Joel P. Martin is with Lockheed Martin Corporation / Army Research Lab, Aberdeen Proving Ground, MD 21005. E-Mail: joel.martin5@us.army.mil.*

fective user study methodologies, we conducted a 2D flow visualization user study, which builds on Laidlaw et al.'s pioneering work [4] but features new strategies and accordingly important improvements. By minimizing bias, along with refining the statistical data analysis, we made this user study as objective as possible and obtained reliable findings from the results. The contributions of our work lie in:

1. *Explicit flow synthesis*. We propose to use explicit/parameterized flow synthesis to combat data-related bias. We developed one such flow synthesizer to automatically generate many flows with nearly the same topological complexity but with different structures. In particular, symmetric flows allowed us to devise novel flow analysis tasks such as symmetric pattern categorization.
2. *Implicit task design*. To reduce task-related bias, we present implicit task design, i.e., designing sample-free pattern-based flow analysis tasks whose fulfillment *indirectly* requires participants to be highly engaged in examining flow directions — the fundamental information governing various complex flow features. Previous work [4] does not address this concept or its importance and hence not all tasks are implicit. In this user study, we designed a *full* set of implicit tasks to minimize bias while supporting sophisticated flow analysis.
3. *Diverse evaluation perspectives*. Despite an inevitable limit on the number of techniques under evaluation, we selected a set that allows us to perform evaluation from diverse perspectives including representation continuity, visual intuition, image contrast, and color mapping. Specifically, grid-based variable-size arrows, evenly spaced streamlines, and LIC variants (basic, oriented, and enhanced versions) are representative of many geometry-based and texture-based 2D techniques in these aspects, apart from the working mechanism.
4. *Hybrid timing strategy*. By characterizing flow analysis tasks, we integrated two timing schemes to collect performance evaluation results: a variable duration for lightweight precision-critical tasks and a fixed duration for heavyweight judgment-intensive tasks. This hybrid strategy helps reveal the subtle differences in visualization effectiveness that may exist between techniques.
5. *Refined statistical analysis*. We refined the statistical data analysis method used in previous work [4]. Outlier results were appropriately processed and the Ryan REGWQ post-hoc homogeneous subset tests were employed to draw valid conclusions.

This paper is organized as follows. Section 2 presents the particulars of the synthetic flow datasets, the selected flow visualization techniques, and the pattern-based task design in our user study. Section 3 describes our session-based test strategy. Experimental results and statistical data analysis are given in section 4. We conclude this paper with a brief summary and outlook on future work.

2 Experimental Components

In this section, we present our approach for three fundamental components of a typical flow visualization user study, i.e., the flow datasets, the flow visualization techniques, and the flow analysis tasks. In general, flow datasets are visualized using various techniques to create images, which are shown to participants to answer questions by performing flow analysis tasks. Statistics on both answer correctness and response time are recorded to determine which techniques are best in which aspects. We can synthesize a wide variety of symmetric flows with an appropriate degree of topological complexity to reduce data-related bias. In addition, our pattern-based task design mechanism engages participants in thorough flow investigation without the bias that sample-based task design may cause. The ultimate goal was to objectively evaluate a set of representative techniques on their effectiveness in 2D flow visualization.

2.1 Synthetic Flow Datasets

To conduct a flow visualization user study, participants are usually asked to examine a collection of images produced using the techniques being evaluated. The use of a single dataset would introduce a learning effect. Thus it is mandatory that different datasets be employed in order to avoid memory retention issues. On the other hand, using multiple flows may incur data-dependent bias as the evaluation is affected by the differing complexities of the flows. Fortunately, this kind of bias can be suppressed to an acceptable degree by generating equally complex synthetic flows. Laidlaw et al. [4] adopted an implicit flow synthesis method, by which nine positions are selected in a unit square and the associated vectors are given within $[-1, 1] \times [-1, 1]$, both using uniform random distribution, before vector interpolation is applied between these initial positions to yield a flow field defined on a specified number of grid points. The topology of the flow [11], [12], [13] resulting from this procedure is unpredictable. In other words, it is difficult to control the number of critical points, their locations, their types, and the overall complexity. Our user study requires a method that can generate centers and symmetric flow patterns, of which the latter are used to design symmetry-categorization tasks. To address this need, we propose to use an explicit flow synthesis strategy, which provides flexibility and control in creating pseudo flows by means of parameterized placement and configuration of critical points.

Zhang et al. [14] gave an informative survey of vector/flow field design and presented a tri-stage method, i.e., initialization, analysis, and editing, for advanced interactive flow synthesis on 2D manifolds. The initialization stage is based on van Wijk's basis vector field approach [8], by which some control parameters are assigned to each critical point to define a basis field and multiple basis fields are combined to create an entire synthetic flow. We employed such a basis vector field scheme while incorporating it with several symmetric pattern templates to allow for fast batch-mode generation of many flows. Compared to the analysis and editing stages of Zhang et al.'s method [14], these pattern templates

serve as some specifically-devised constraints on both the distribution and the configuration of critical points to achieve automatic flow design. Thus our explicit/parameterized flow synthesis strategy is capable of maintaining nearly the same degree of topological complexity among the resulting flows, effectively reducing data-related bias. To our knowledge, our flow visualization user study is the first to value and apply *explicit* flow synthesis.

The positions and types of critical points largely determine the structure and behavior of a flow field. Laidlaw et al. [4] employed saddles, foci, and nodes as the constituent elements in their task design, whereas we use saddles, foci, and centers because it may be difficult for participants to distinguish between foci and nodes, since their actual shapes may significantly differ from ideal illustrations. Our flow synthesizer is built on the explicit specification of centers/foci coupled with the derivation of saddles (from the interaction among centers and foci). It enables not only accurate configuration of centers/foci to meet the need of our task design but also more control over the number of critical points.

To build a basis vector field, a force composition approach is adopted in our flow synthesizer, which governs the influence of an *e*xplicitly *s*pecified *c*ritical *p*oint (ESCP), either a center or a focus, on an arbitrarily placed virtual massless particle. The parameters for an ESCP are radial force *RAD* (0 for centers), rotational force *ROT*, clockwise/counter-clockwise orientation, sink/source (i.e., attracting/repelling for foci), and a scaling variable λ that is used in combination with the inverse squared distance to emulate force attenuation effects. Given these parameters, the two components of the composite force on the particle can be analytically represented. The net force of multiple ESCPs on an arbitrary particle is obtained through a linear composition with the weight of each ESCP inversely proportional to the particle-ESCP distance. Despite the inconsistency with Newton's second law in the strict sense, the use of a vector with the two components proportional to the two components of the net force, respectively, effectively models the motion direction and velocity magnitude of the particle.

The ability of our flow synthesizer to explicitly place and configure centers/foci makes it possible to create symmetric flows in support of our pure pattern-based task design mechanism. To increase the diversity of symmetric flow patterns, ESCPs are randomly placed on a per pair basis in a unit square, which is then sampled at two resolutions. One resolution *TRES* is used for texture-based visualization techniques and the other, *GRES*, for geometry-based OpenGL-dependent techniques. To prevent ESCPs from being either crowded together or separated excessively, three ESCP templates are constructed via a polar coordinate system to specify both the distance range ρ in which each ESCP is radially jittered from the center of the square and the angle range ω in which each ESCP is circularly jittered off of one of K (an even integer) uniformly distributed center-based reference radial lines (Fig. 1). The combined effect of radial jittering and circular jittering leads to random distribution of ESCPs, one within each of the truncated-fan-shaped blocks. In this way, a wide range of flow fields can be synthesized, each with a relatively balanced layout of a fixed number of ESCPs and a slightly varying number of derived saddles to maintain similar topological complexity. In fact, random placement is performed on only one half of the K ESCPs, i.e., *the primary ESCPs* (one within each light blue block in Fig. 1), whereas the other half, consistent in the center/focus type with the primary, are placed precisely by exploiting the specific spatially symmetric property and are hence called *the mirror ESCPs* (one within each gray block in Fig. 1). Likewise, the sink/source type and the clockwise/counter-clockwise orientation are randomly selected for each primary ESCP, whereas both are accordingly determined for its mirror ESCP. Specifically, two paired ESCPs are consistent except for the clockwise/counter-clockwise orientation for x-axis and y-axis symmetric flows. In fact, multiple asymmetric flows can be produced by altering the sink/source type and/or the clockwise/counter-clockwise orientation of the mirror ESCPs of a symmetric flow. Topologically asymmetric flows, formed by geometrically symmetric ESCPs, allow for thorough evaluation of competitive visualization techniques by challenging the participant's perception of the flow both locally (for the flow direction and the types of every critical point) and globally (for the position of every critical point and the consistency/inconsistency between each pair of ESCPs in the types).

The parameters for the explicit flow synthesizer in our user study are: $RAD = 0$ for centers and 200 for foci, $ROT = 1000$, $\lambda = 0.125$, $K = 8$, $\rho = [0.24, 0.42]$, $\omega = [-7.5, 7.5]$, $TRES = 706$, and $GRES = 400$. These values for the first six parameters can generate high-quality flow fields. The ratio between *RAD* and *ROT* guarantees that centers and foci are easily distinguishable in an image if the visualization technique itself is able to show both of these two kinds of flow features. For x-axis/y-axis symmetric flows, no ESCP is put at the square center, whereas an ESCP of type center may be optionally placed there to make center-symmetric flows. Tests with the above settings indicate that the number of all critical points in a synthetic flow falls within the range [11, 15].

2.2 Flow Visualization Techniques

A flow visualization user study is usually concerned with flow direction, flow orientation, and velocity magnitude. In our user study, flow direction is twofold [15], [16], denoting both negative and positive directions tangent to the flow. However, flow orientation refers to only the positive direction of the flow, as is the case with the definition of oriented LIC [16]. Furthermore, these three flow characteristics may be examined indirectly (*implicitly*) by designing pattern-based tasks related to flow topology.

We considered 54 techniques, which are categorized into hedgehogs, streamlines, and LIC (Table 1). The selection of an appropriate set from these candidates was based on thorough investigation. First, this set contains traditional methods, popular approaches [2], [16], [17], and recent advances [18] in 2D flow visualization such that conclusions may be drawn as to *whether* the more

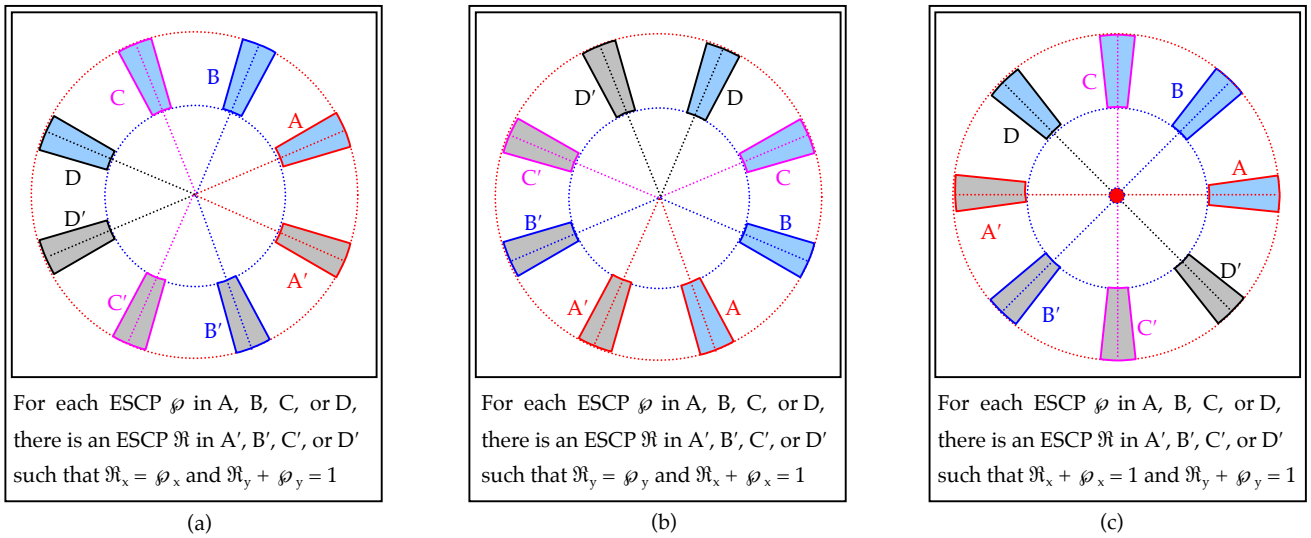


Fig. 1. The three ESCP templates employed in our flow synthesizer to define (a) four pairs of x-axis symmetric ESCP blocks, (b) four pairs of y-axis symmetric ESCP blocks, and (c) four pairs of center-symmetric ESCP blocks, respectively, for diverse but relatively balanced placement of eight ESCPs in a unit square [0, 1] × [0, 1]. Each primary ESCP is jittered within a light blue block while the mirror ESCP, consistent in the center/focus type with the primary, is precisely placed in the opposite gray block based on the specific symmetry.

recent techniques are better. Second, each technique conveys at least two flow characteristics. Comparisons may be made to determine *which* techniques outperform the others in *which* aspects. Third, this set addresses geometry-based and texture-based methods, exhibiting a transition in the degree of continuity (0D, 1.5D, and 2D) regarding flow representation. This coverage may help find clues for *why* some techniques are more effective than others. Fourth, each technique adopts a color map such that hints may be gained on *how* to exploit the strengths while conquering the weaknesses in the use of color. Color encoding involves visual perception and human cognition issues and is an emerging research area [19] of visualization. Our focus is not placed on the design or selection of the best color map schemes. Instead the goal of our work in this aspect is to evaluate visualization techniques equipped with a colorwheel [20] or rainbow color map and report their respective advantages and disadvantages.

Table 2 gives a brief description of the seven selected techniques, the flow attributes that each conveys, and the flow analysis tasks that each supports. Fig. 2 shows the images produced by using each technique to visualize the same synthetic x-axis symmetric flow, as well as the two color maps.

These seven techniques have each been acclaimed to be effective for visualizing 2D flows. Compared to the pin-like glyph (top left in Table 1) and the intensity-tapering icon (top right in Table 1), the arrow shape is relatively intuitive and unambiguous in displaying the flow orientation. Variable-size arrows emulate some degree of random distribution, alleviating the cluttering problem that usually occurs with fixed-size arrows in turbulent flow areas. Grid-based arrows outperform jittered ones (i.e., actual random distribution) in facilitating mental reconstruction of directional information [6]. An uncontrolled layout of streamlines tends to incur cavities or cluttering, whereas an evenly spaced streamline placement [18], [21], [22] can produce an aesthetic as well as informative image. The colorwheel [20] maps two characteristics of the flow (orientation and magnitude) to the three components of HSV color space (hue, saturation, and value/brightness in Fig. 2i). Preliminary candidates-screening tests showed that for arrows and evenly spaced streamlines, the use of either white primitives over a colorwheel background or rainbow color-mapped primitives over a black background is more understandable than the use of black primitives over a white background in depicting the velocity magnitude. One hypothesis is that the colorwheel might help with visual interpolation across the flow in an image of arrows or streamlines, but the preliminary tests indicated that this does not apply to the texture-based LIC variants. The indirect orientation representation offered by the colorwheel seemed to add confusion to LIC images.

Although the seven techniques may not be the state of the art in flow visualization, they are representative of many geometry-based and texture-based methods in important evaluation aspects such as representation continuity, visual intuition, image contrast, and color mapping. For example, IBFV [8] is newer than the three LIC variants that we selected and is famous for its novelty, simplicity, versatility, and performance. Except for these advantages that are invisible in a flow visualization user study, basic IBFV and enhanced IBFV (with a high-pass filtering post-process) are similar to BasicLIC and EnhancedLIC, respectively, in the aforementioned evaluation aspects. This is also the case with many other texture-based techniques [1].

TABLE 1
THE 54 CANDIDATE TECHNIQUES WE CONSIDERED, FROM WHICH A SMALL SET WAS CHOSEN FOR THE USER STUDY

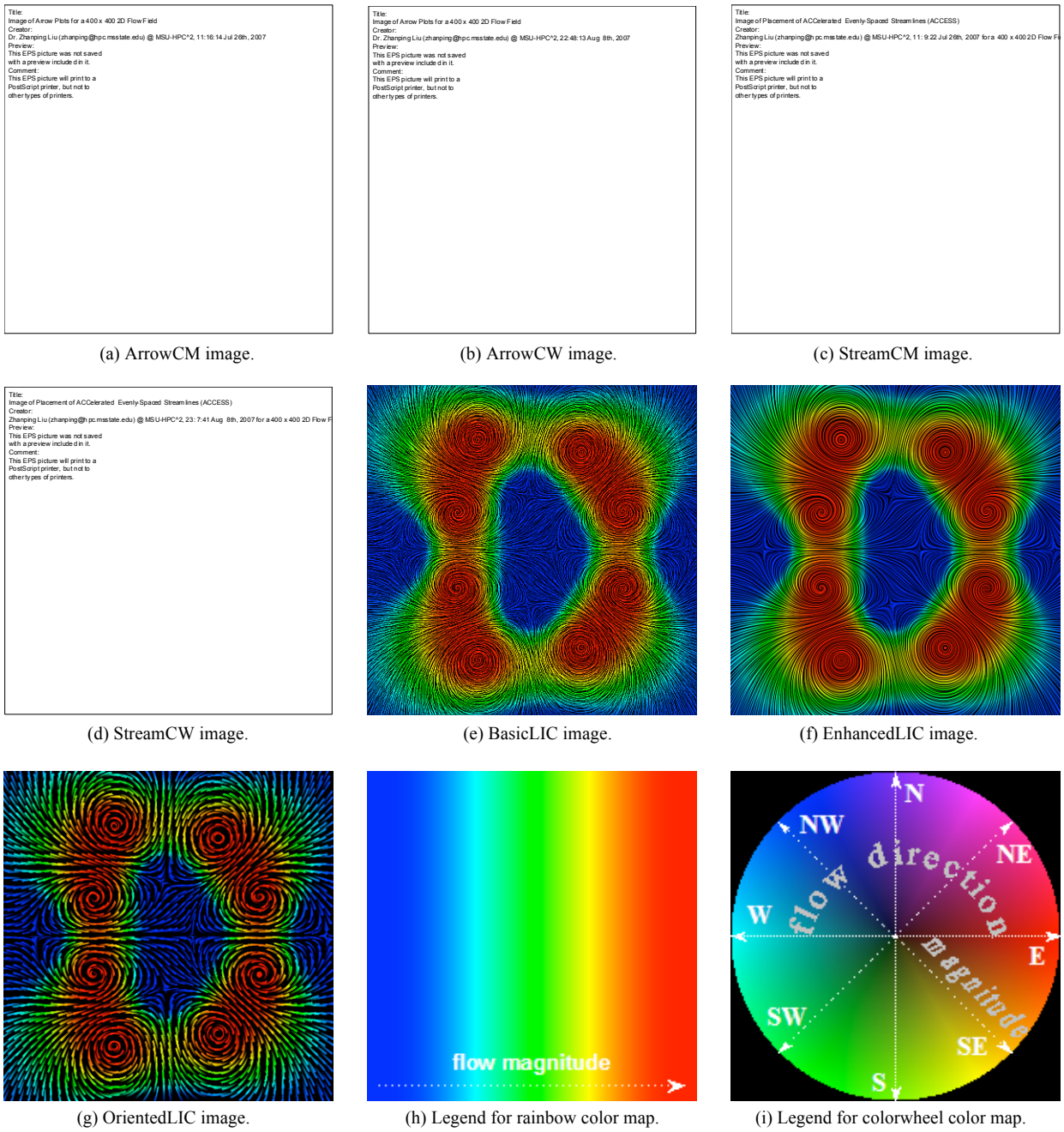
Hedgehogs (0D)	**type**			
	size	fixed		variable: magnitude-based map
	layout	at regular grid points		at jittered grid points
	color	white over colorwheel background	black over white background	rainbow color map over black
Streamlines (1.5D)	**type**	seeds at regular grid points	seeds at jittered grid points	evenly spaced placement
	color	white over colorwheel background	black over white background	rainbow color map over black
LIC (2D)	**type**	basic LIC	oriented LIC	enhanced LIC
	color	colorwheel color map	gray-scale	rainbow color map

* The candidate techniques of each family (hedgehogs, streamlines, or LIC) are derived from all possible combinations across the family's attributes (e.g., type, size, layout, and color for the hedgehogs family). The three families have 36 (= 3 × 2 × 2 × 3), 9 (= 3 × 3), and 9 (= 3 × 3) candidate techniques, respectively.

TABLE 2
THE SEVEN TECHNIQUES SELECTED FOR OUR FLOW VISUALIZATION USER STUDY

Visualization Techniques Selected		Flow Attributes Conveyed	Analysis Tasks Supported
ArrowCM	arrows placed at regular grid points variable size based on magnitude rainbow color map by magnitude over black background	flow direction flow orientation velocity magnitude	locate critical points classify saddles hard to distinguish circles from foci
ArrowCW	arrows placed at regular grid points variable size based on magnitude white over colorwheel background	flow direction flow orientation velocity magnitude	locate critical points classify saddles hard to distinguish circles from foci
StreamCM	evenly spaced streamlines rainbow color map by magnitude over black background	flow direction velocity magnitude	locate critical points classify all critical points distinguish circles from foci
StreamCW	evenly spaced streamlines white over colorwheel background	flow direction flow orientation velocity magnitude	locate critical points classify all critical points distinguish circles from foci
BasicLIC	basic LIC rainbow color map by magnitude	flow direction velocity magnitude	locate critical points classify all critical points distinguish circles from foci
EnhancedLIC	enhanced LIC rainbow color map by magnitude	flow direction velocity magnitude	locate critical points classify all critical points distinguish circles from foci
OrientedLIC	oriented LIC orientation by the increase in intensity rainbow color map by magnitude	flow direction flow orientation velocity magnitude	locate critical points classify all critical points distinguish circles from foci

For the seven techniques, we conducted iterative internal tests (involving six visualization experts) to tune their parameters for optimal visualization results. The settings chosen for our formal user study are as follows. Each synthetic unit-square flow is sampled at two resolutions (Section 2.1), i.e., $TRES = 706$ for BasicLIC, EnhancedLIC, and OrientedLIC and $GRES = 400$ for ArrowCM, ArrowCW, StreamCM, and StreamCW. Graphical primitives generated using each of the latter four for a 400 × 400 flow dataset are rendered in anti-aliasing mode to a 706 × 706 OpenGL view such that the seven techniques have the same output image size. For each flow dataset, the velocity magnitude is transformed to [1.0, 1000.0] via histogram equalization before the colorwheel or rainbow color map is applied. The rainbow scheme (Fig. 2h) linearly maps the lowest magnitude to blue and the highest to red. The colorwheel mode (Fig. 2i) linearly maps velocity magnitude to brightness within [0.2, 1.0] and to saturation within [0.4, 1.0]. In addition, it maps flow orientation to hue, with red assigned to vector (1.0, 0.0) — the starting radial line. The colormaps we employed, though not perceptually linear, are based upon mappings that are most commonly used by application scientists and upon visualization system defaults. For ArrowCM and ArrowCW, a uniformly spaced lattice of 45 × 45 (out of 400 × 400) grid points is created through 9 × 9 sub-sampling to place arrows. The variable arrow size (in cells) obtained by 1.2 × [2.0 + log(magnitude)] falls within [2.4, 10.8], with the arrow head fixed to 2.4 in length and width. StreamCM and StreamCW, built on the ADVESS algo-



(a) ArrowCM image. (b) ArrowCW image. (c) StreamCM image.

(d) StreamCW image. (e) BasicLIC image. (f) EnhancedLIC image.

(g) OrientedLIC image. (h) Legend for rainbow color map. (i) Legend for colorwheel color map.

Fig. 2. The images (a-g) generated by using each of the seven selected techniques to visualize a synthetic x-axis symmetric flow and the two color map schemes (h-i) used.

rithm [18], are configured to generate 1.0% density evenly spaced streamlines. BasicLIC and EnhancedLIC adopt a 15-pixel-wide box kernel and white noise. In particular, EnhancedLIC employs two LIC iterations followed by 3 × 3 Laplace high-pass filtering [17]. OrientedLIC uses a 10-pixel-wide ramp kernel and sparse noise that is synthesized by jittering 3 × 3 white crosses within uniformly distributed 9 × 9 black blocks of a 706 × 706 quad.

2.3 Flow Analysis Tasks

Given a collection of images generated using the seven techniques for a set of synthetic flows, the techniques can be evaluated by asking participants to conduct a series of tasks. Thus the performance of an *average* participant in visual flow analysis reflects the effectiveness of the technique being used.

Besides critical point recognition and classification, a sample-based flow analysis task was devised by Laidlaw et al. [4] in their user study. The participant was shown a randomly placed circle (of which the center is hence a random sample) and asked to click on the point along the circle that a particle advected from the center is to hit. Since the complexity of a flow usually varies with location, it is more difficult to accomplish this task in turbulent areas than in laminar areas and the selection of the circle's radius may further compound this issue. In addi-

tion, the user's ability to point and click accurately with a mouse, irrelevant of the perception, analysis, and judgment, affects the test result. On the other hand, critical point recognition is an implicit, pattern-based, high-level task because the associated flow exploration process considers the whole field instead of a single point to detect patterns of interest. This is also the case with critical point classification that matches topological templates (patterns) to the flow structure exhibited in the area around a critical point. These two pattern-based tasks can be presented to the participant in a simple form, while they implicitly require that the participant thoroughly inspect the flow direction across the whole domain (globally) and around an area of interest/features (locally), respectively, suppressing task-related bias.

Implicit task design is a new concept that we propose to use in flow visualization user studies to deal with flow analysis bias. As a vector attribute, flow direction provides the fundamental information that distinguishes a flow field from a scalar field, allowing us to define, recognize, and interpret many flow features or patterns (e.g., critical points, separatrices, and periodic orbits). In this sense flow direction is much more important than the associated scalar quantities of the flow such as velocity magnitude, pressure, temperature, and other derived attributes. In fact, research on flow visualization has been primarily focused on ways of displaying flow direction, because how well a technique delineates the *general*, directional information largely governs its effectiveness in conveying *specific* flow features. With the core objective being sample-free flow analysis to overcome bias, implicit task design usually includes the direct use of specific well-known flow features (e.g., critical point recognition and critical point classification) and in-depth flow structures (e.g., identification of separatrices and periodic orbits) — *real tasks*. It may also work by creating some appropriate 'synthetic' pattern-based tasks, analogous to generating synthetic flows. While synthetic flows are useful for minimizing data-related bias (originating from different topological complexities) and avoiding potential size problems with real datasets, these *synthetic tasks* are intended to reduce task-related bias (resulting from flow sampling and point-and-click operations) and to relieve non-expert participants from understanding complex, possibly domain-specific details. Synthetic tasks are pattern-based, involving easy-to-understand but challenging questions and requiring intensive analysis of flow directions. It is worth mentioning that synthetic tasks are complementary to real tasks and these two types constitute implicit task design. Motivated by the necessity for and significance of implicit task design, we present a synthetic task that was integrated with two real tasks, i.e., critical point recognition and classification, in our user study.

Our explicit, template-based flow synthesizer (section 2.1) enables us to design a synthetic pattern-based flow analysis task, i.e., symmetric pattern categorization. This task does not require precise point-and-click operations. However, it is challenging in that the participant needs to be highly engaged in examining the flow direction both globally and locally in an effort to determine whether the entire pattern is x-axis symmetric, y-axis symmetric, center-symmetric, or asymmetric, both geometrically and topologically. Merely locating all critical points is insufficient to fully understand a flow field. Instead the participant has to observe the flow direction around the critical points to determine if two local patterns under consideration are symmetric. For example, some topologically asymmetric flows made up of geometrically symmetric critical points are very easily mistaken to be symmetric. In fact, a similar challenge applies to the case in which the participant is shown a flow to choose between three symmetry types due to the relatively balanced distribution of ESCPs (Section 2.1). Thus visualization techniques can be evaluated in terms of their effectiveness as the participant is indirectly "forced" to study the flow direction in a global+local fashion.

Table 3 lists the three pattern-based tasks of our user study, i.e., *c*ritical *p*oint *r*ecognition (CPR), *c*ritical *p*oint *c*lassification (CPC), and *s*ymmetric *p*attern *c*ategorization (SPC). Apart from SPC, there are many synthetic tasks that may be used in a flow visualization user study. For example, two or three critical points (centers, foci, and saddles) can be combined with a variety of configurations to define some *c*omposite *t*emplates (CT) in support of CT-based CPR-like pattern recognition and CPC-like pattern classification. Other synthetic tasks include checking if flows A and B have a CT pattern in common, judging if flow A is a rotational version of flow B, and determining if flow A is exactly part of flow B.

3 Test Strategy

In this section we present a session-based test strategy adopted in our flow visualization user study. Given N x-

TABLE 3
THREE PATTERN-BASED TASKS USED IN OUR USER STUDY

Pattern Questions			Participant Operations	Analysis Tasks	Investigation Scopes
CPR	Where are critical points?		mouse click on points in an image	pattern recognition	global
CPC	Is the critical point	a saddle?	mouse click on a radio button	pattern classification	local
		a center?			
		a focus?			
SPC	Is the overall flow pattern	x-axis symmetric?	mouse click on a radio button	pattern recognition pattern classification pattern comparison	global + local
		y-axis symmetric?			
		center-symmetric?			
		asymmetric (optional)?			

axis symmetric, N y-axis symmetric, N center-symmetric, and optionally N asymmetric flows, we visualize these M (M = $3N$ or $4N$) datasets using the seven techniques to produce $7M$ images. Each flow is synthesized with the parameters and templates discussed in Section 2.1, which define the location and type (either center or focus) of every ESCP and the symmetry type of the entire pattern. The design of the data generator guarantees that no nodes exist in any synthetic flow. Saddles, derived from the interaction among the ESCPs, can be detected (and located) through a Newton-Raphson root-finding method. These explicit and extracted attributes (i.e., the aforementioned locations and types) provide the ground truth of a flow.

In general, the effectiveness of a visualization technique is determined by answer correctness and response time. A more effective technique allows a user to get a correct answer faster. Likewise, given a fixed amount of time, more correct answers tend to result from a more effective technique than from a less effective one. Thus we propose to use *a variable-duration test scheme* for CPR, but *a fixed-duration test scheme* for both CPC and SPC. In the former case, mouse click positions and response time are recorded in a *session*. In the latter case, as many tasks as possible are presented to the participant one by one within 30 seconds and only radio-button choices are recorded in a session. The variable-duration scheme seeks to "curb" the participant during relatively quick (considering the recognition of a single critical point) precision-critical (yet with an error tolerance, Section 4.1) flow analysis, whereas the fixed-duration scheme is intended to "push" the participant during relatively slow judgment-intensive flow analysis.

The session concept integrates different task management schemes into task delivery. A session may contain one CPR task (for recognizing all critical points), up to 30 CPC tasks, or up to 40 SPC tasks. A *CPR session*, a *CPC session*, and an *SPC session* that are used to evaluate the same technique make up a *set*. Seven sets, one for each technique, constitute a *cycle*. Each participant goes through three cycles, with 63 sessions presented in random order. The 21 CPR sessions are built on a random selection of 21 images produced by using each of the seven techniques thrice. Each of the 21 CPC sessions is constructed by randomly choosing 30 images that are generated using the technique under evaluation. Thirty circles are drawn over the 30 images, respectively, to mark 10 randomly selected saddles, 10 randomly selected centers, and 10 randomly selected foci. These images are shown to the participant in random order. Each of the 21 SPC sessions is created by randomly selecting 40 images that are produced using the technique under evaluation to visualize 10 x-axis symmetric, 10 y-axis symmetric, 10 center-symmetric, and 10 asymmetric flows. These 40 images are shown to the participant also in random order. As an option, asymmetric flow images may be excluded to make 30-task SPC sessions. The user study reported in this paper did not use asymmetric flows to simplify statistical analysis of the result. However, categorizing the symmetric patterns proved to be a challenging task (Section 4.3).

We developed a batch mode tool called *TestGen* for implementing the session-based test strategy and a *flow visualization user study system* (FlowVUSS) for executing the strategy. TestGen is provided for the experimenter to initialize a test, while FlowVUSS runs in a training mode or a test mode (Figures 3-5) for the participant to perform flow analysis tasks. The test procedure we designed is as follows. First the participant is given a brief introduction to the goal of the test, the background of flow visualization, necessary knowledge about flow topology, and the three kinds of flow analysis tasks. Next FlowVUSS runs in the training mode with a sequence of pre-designed task sessions to familiarize the participant with the user interface and tasks. The submission of any answer is followed by immediate feedback with the correct answer. Once the training stage is over, TestGen is used to create a test file containing 63 randomly generated and randomly scheduled sessions of tasks. Then FlowVUSS runs in the test mode and accesses the test file to present the task sessions to the participant. The answers and response time of the participant are written to an output file for the subsequent statistical analysis.

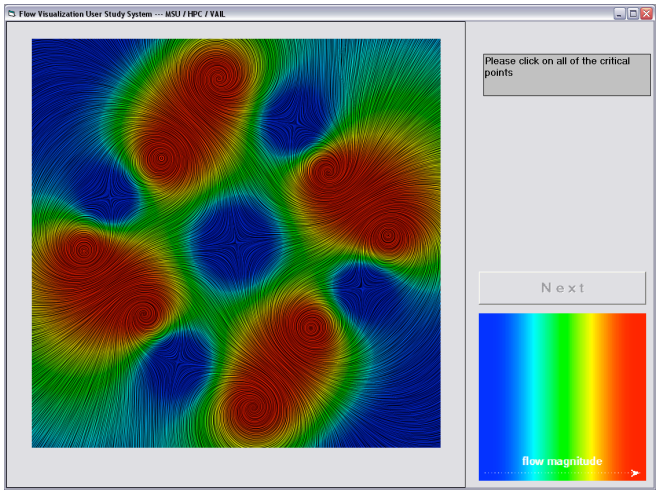


Fig. 3. FlowVUSS running in the test mode with an EnhancedLIC-based CPR session.

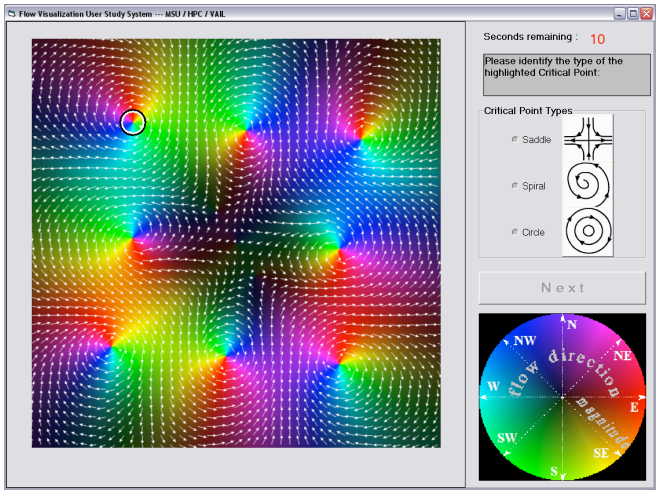


Fig. 4. FlowVUSS running in the test mode with an ArrowCW-based CPC session.

4 Results and Discussions

We recruited four CFD experts and 16 graduate students in science and engineering disciplines. Each non-employee participant was compensated with $10. Our user study did not compare expert and non-expert participants in flow analysis performance since it has been reported [4] that in general these two groups do not exhibit a statistically significant difference. Any difference would be mitigated by the introduction to flow visualization and the training sessions given prior to each formal test. Thus this user study was primarily focused on a quantitative comparison between the seven techniques in visualization effectiveness. A typical approach is to obtain the means and standard errors of some dependent measures (e.g., response time) of interest across multiple conditions (e.g., various visualization techniques) to observe the influence that each variable has on the measures [23], [24].

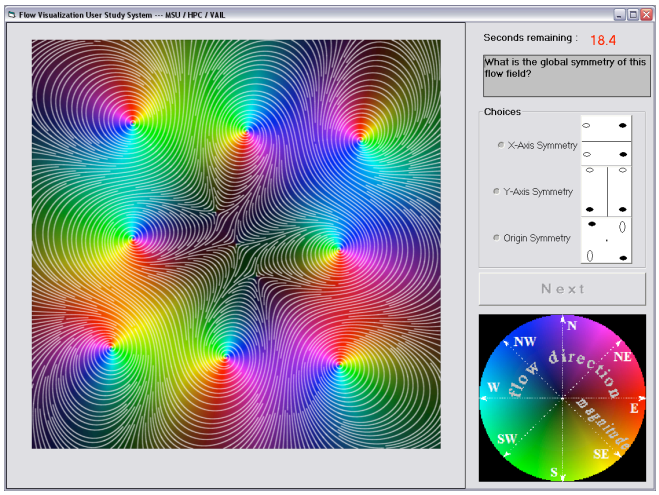


Fig. 5. FlowVUSS running in the test mode with a StreamCW-based SPC session.

The participants of our study performed a total of 5079 CPR trials, 7467 CPC trials, and 4948 SPC trials. We calculated an error statistic for each type of trials. In addition, we recorded recognition time and location error from the CPR trials, while we collected classification/categorization time from the CPC and SPC trials. We first examined histograms to summarize these dependent measures; as expected, the response time and the location error showed skewed normal distributions. We then determined outliers on a case-by-case basis, by investigating the tails of the distributions and noting values that appeared after conspicuous gaps in the histogram. As recommended by Barnett and Lewis [25], we replaced each outlier with the median of the responses in the experiemental cell. Table 4 summarizes the dependent measures that we collected and calculated, and describes the outliers that we found for each measure.

We analyzed the data using chi-square tests and univariate analysis of variance (ANOVA). For the ANOVA, we modeled our experiment as a repeated-measures design that considers *participant* as a random variable and all other independent variables as fixed. In addition, since we are primarily interested in comparing the performance of the seven different visualization techniques, we also calculated post-hoc homogeneous subsets using the Ryan REGWO test [26].

Using our understanding of the seven flow visualization techniques and the comments/input from the participants, we postulate the underlying reasons for the results that we obtained. The absolute differences in response time for CPR/CPC/SPC turned out to be small in our user study, regardless of the statistical differences. Thus we assign a higher priority to correctness than speed to provide *c*orrectness-*o*ver-*s*peed-sorting (COSS) when evaluating the seven techniques in *overall* effectiveness. Less important than these two measures, precision (applicable to CPR only) is reported for supplemental analysis.

4.1 Critical Point Recognition (CPR)

We recorded response time, recognition error, and location error during CPR sessions (each with many critical points to be found in an image). The response time means how long it takes to recognize a critical point. A critical point is (properly) recognized if and only if there is a mouse click within a threshold radius (e.g., 35 pixels based on our flow synthesizer), with any duplication rejected. A recognition error refers to either a false negative (negligence) or a false positive (mis-identification) occurrence. The absolute error, a non-negative integer, denotes the number of neglected critical points plus number of invalid mouse clicks (including false positives and duplicate recognitions). The location error, associated with a correct recognition, is defined as the distance (in pixels) between the critical point and the mouse click position.

Figures 6-7 show the absolute recognition error and mean response time, respectively, for each of the seven techniques. In correctly recognizing critical points, EnhancedLIC was the most effective, followed by StreamCM, BasicLIC, OrientedLIC, StreamCW, Ar-

TABLE 4
THE DEPENDENT MEASURES THAT WE COLLECTED AND CALCULATED

Measure	*N*	Outlier Definition	Number of Outliers	Outlier Percentage	Figure
CPR Error	143	—	—	—	6
CPR Recognition Time	5079	> 10 seconds	45	0.9%	7
CPR Location Error	5079	> 35 pixels	68	1.3%	8
CPC Error	753	—	—	—	9
CPC ClassificationTime	7467	> 20 seconds	5	0.07%	10
SPC Error	323	—	—	—	11
SPC Categorization Time	4948	> 25 seconds	3	0.06%	12

rowCM, and ArrowCW. This order indicates that StreamCM even outperformed two texture-based techniques BasicLIC and OrientedLIC, whereas StreamCW fell behind these two LIC variants apparently due to the use of a different color map. ArrowCM and ArrowCW were largely inferior to the other five and reiterated the influence of color mapping on geometry-based techniques. Regarding response time, no statistical differences existed either between StreamCM, BasicLIC, OrientedLIC, and EnhancedLIC or between ArrowCM and StreamCW, while the former four allowed for faster user response than the latter two. ArrowCW required the longest time for users to recognize a critical point. The loose consistency between the order in correctness and that in speed demonstrates that fewer recognition errors were actually not due to more time consumption. Instead it is the more effective flow representation of a technique that allows users to respond to CPR more quickly and more correctly. Strengthened by this loose consistency, our COSS rule sorts the seven techniques by CPR effectiveness in decreasing order as

- EnhancedLIC, StreamCM, BasicLIC, OrientedLIC, StreamCW, ArrowCM, ArrowCW.

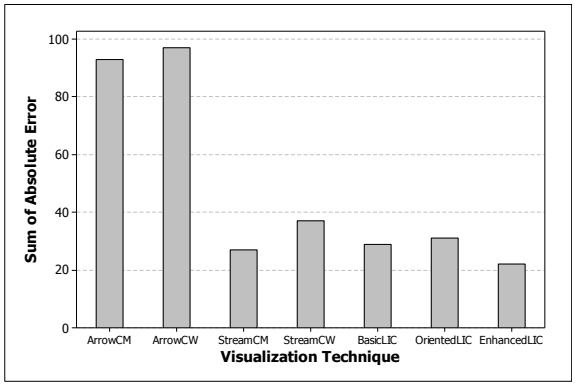


Fig. 6. Number of CPR errors for each visualization technique (N = 143). EnhancedLIC was the most effective in correctly recognizing critical points. The error count was different for each visualization technique ($\chi^2(6) = 132$, $p < 0.001$).

EnhancedLIC was the most effective for the dense representation with clear-cut flow streaks produced by LIC iterations and high-pass filtering. The 2D continuity enables intuitive perception of the flow without visual interpolation. Sharp yet thin flow streaks provide strong direction cueing to expose topological elements. BasicLIC was inferior to EnhancedLIC since the blurring effect of low-pass filtering hinders participants from finding critical points. OrientedLIC was less effective than EnhancedLIC because the decrease in density of flow depiction demands visual interpolation and because 3D-like, thick, intensity-tapering flow streaks pose distractions for discerning saddle points. With these weaknesses, OrientedLIC was even slightly less effective than BasicLIC, and both were a little bit worse than StreamCM. Although StreamCM is a geometry-based technique with 1.5D continuity, an evenly spaced streamline placement provides an organized informative representation, without cluttering or loss of information, to facilitate visual interpolation across the flow. The ADVESS algorithm [18] underlying StreamCM (and StreamCW) is capable of minimizing cavities around critical points to highlight salient features. StreamCM offers relatively straightforward direction cueing and clean feature display, whereas BasicLIC requires more direction extraction and topology reconstruction from a blurry LIC texture. Although the rainbow color map is visually nonlinear, its use in StreamCM provides some degree of spatial correlation between flow direction and velocity magnitude, as is the case with the LIC variants. The local correlation tends to lend itself to the identification of critical points. This effect is particularly helpful in StreamCM as the black background creates high image contrast to make flow features easily discernible.

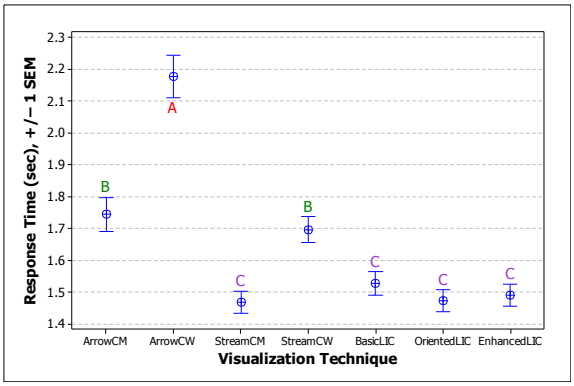


Fig. 7. Mean time (in seconds) to recognize a critical point (N = 5079). There was a main effect of visualization technique on the time to recognize a critical point ($F(6,115.3) = 19.9$, $p < 0.001$). Means with the same letter are not significantly different at $p \leq 0.05$ (Ryan REGWQ post-hoc homogeneous subset test).

One major goal of including StreamCW and ArrowCW in the user study was to explore the effectiveness of equipping geometry-based techniques with a color map background to emulate a dense representation. The colorwheel scheme establishes a one-to-one map between the hue of the background and the flow orientation. We assumed this map would aid in visual interpolation across lines or arrows to create a continuous impression of the flow, though StreamCW fell significantly behind the LIC variants and even StreamCM in CPR. The comparison between StreamCM and StreamCW indicates that the colorwheel scheme was less effective than the rainbow color map. This result was probably due to the rare use, complex legend, and low contrast of the colorwheel, and even worse, due to the visual distraction that affects CPR at a global scale. Similar findings and reasons hold when we compare ArrowCM and ArrowCW. These two arrow-based techniques were far inferior to the other five in CPR because of 0D continuity.

Fig. 8 shows the mean location error for each technique. StreamCM, StreamCW, BasicLIC, OrientedLIC, and EnhancedLIC performed nearly the same in locating recognized critical points, except that EnhancedLIC was better than BasicLIC due to the large difference in image contrast. The two arrow-based techniques were less accurate than the others since the discreteness both along and across the flow direction hinders visual interpolation. Users achieved more precise location using ArrowCW than using ArrowCM. This result was probably due to the

bright 'dot' that the colorwheel map creates at the core of each center/focus.

4.2 Critical Point Classification (CPC)

We collected response time and classification error during CPC sessions. Under the fixed-duration test scheme, the response time is an average value, equal to 30 seconds divided by the number of CPC tasks completed in a session. A classification error occurs if a wrong type is selected for a critical point (one randomly marked in each image). Figures 9-10 show the number of classification errors and mean response time, respectively, for each of the seven techniques. In correctly classifying critical points, EnhancedLIC was the most effective, followed by StreamCW, which was marginally better than StreamCM. BasicLIC was less effective than StreamCM, but much more effective than OrientedLIC. ArrowCW and ArrowCM performed nearly the same, but much worse than the other five. In terms of time, users responded more quickly with StreamCM and EnhancedLIC than with BasicLIC and OrientedLIC. Compared to StreamCW, StreamCM required less time for a CPC task. ArrowCM and ArrowCW resulted in the slowest user response. Based on the COSS rule, the seven techniques may be sorted by CPC effectiveness in decreasing order as

- EnhancedLIC, StreamCW, StreamCM, BasicLIC, OrientedLIC, ArrowCW, ArrowCM.

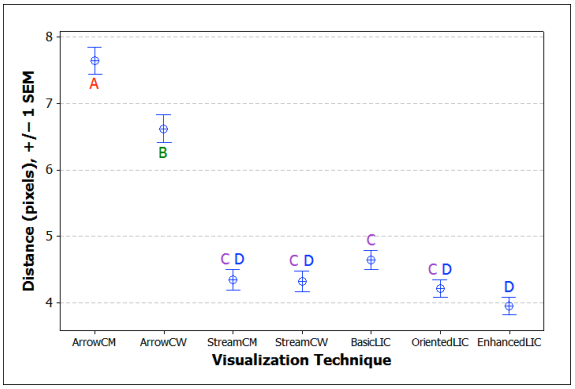


Fig. 8. Mean location error (in pixels) for a recognized critical point (N = 5079). There was a main effect of visualization technique on the mean location error ($F(6,114.8) = 40.1$, $p < 0.001$). Means with the same letter are not significantly different at $p \le 0.05$ (Ryan REGWQ post-hoc homogeneous subset test).

There were 753 classification errors out of 7,467 CPC answers, with about 12% from the misjudgment between saddles and centers/foci. The majority of the 12% stemmed from ArrowCM and ArrowCW as arrows are weak in conveying saddles. The other five techniques produced only a few saddle-versus-center/focus errors, apparently because of the 1.5D/2D continuity and the better distinction between the shape of saddles and centers/foci. Thus our analysis of the CPC effectiveness below is focused on the 88% of the classification errors — the misjudgment between centers and foci. Furthermore, the statistics show that the majority of these errors were caused by the misclassification of centers as foci. Thus the analysis turns into a discussion about the effectiveness of the techniques in conveying centers (closed ellipses), which may be actually visualized as tightly spiraling foci (open curves). Besides numerical accuracy issues (e.g., involved in integration-based techniques), flow field representation plays an important role in depicting centers. Specifically, spatial continuity determines if a center can be entirely delineated. Without enough spatial continuity, even a great deal of visual interpolation may result in a center being misclassified as a tightly spiraling focus. Another factor is visual intuition, meaning how easily the user perceives or recognizes a center. Insufficient intuition poses a visual burden of curve extraction to determine if there is a center embedded in a dense texture.

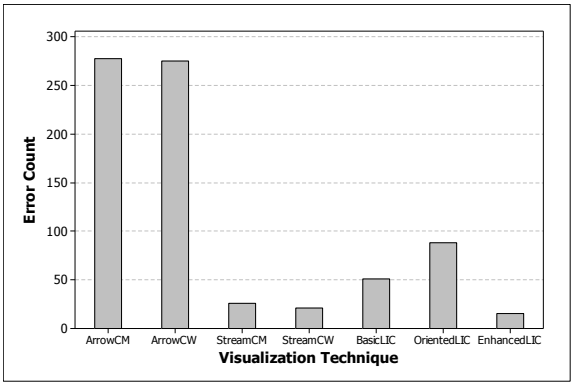


Fig. 9. Number of CPC errors for each visualization technique (N = 753). EnhancedLIC was the best to correctly classify critical points. The error count was different for each visualization technique ($\chi^2(6) = 772$, $p < 0.001$).

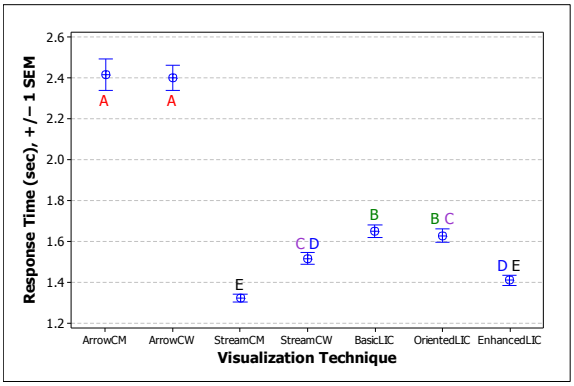


Fig. 10. Mean time (in seconds) to classify a critical point (N = 7467). There was a main effect of visualization technique on the time to classify a critical point ($F(6,116.2) = 30.9$, $p < 0.001$). Means with the same letter are not significantly different at $p \le 0.05$ (Ryan REGWQ post-hoc homogeneous subset test). The streamline and LIC methods exhibited a statistically significant difference from the arrow methods.

With 0D continuity, ArrowCM and ArrowCW performed the worst in visualizing centers. Since the user concentrates on a region of interest during each CPC task, color mapping is less critical than spatial continuity and hence did not make a noticeable difference between these two techniques. This implies that the colorwheel introduces visual distraction globally (for CPR, Fig. 6) more than locally (Fig. 9). As an image-based representation, OrientedLIC exploits the synthesized flow streaks to show some centers and was far more effective than ArrowCM and ArrowCW in the test. However, the use of sparse noise in OrientedLIC produces cavities, compro-

mising 2D continuity. As a result, OrientedLIC was less effective than BasicLIC since the latter achieves real 2D continuity (Fig. 9). The ADVESS algorithm [18], [27] underlying StreamCM and StreamCW provides robust detection of centers and places at least one closed streamline around each center. This capability, along with 1.5D continuity and the better visual intuition, appears to make StreamCM and StreamCW more effective than BasicLIC, which suffers from blurring issues. Color mapping seemed to make a tiny difference, as StreamCW was a little bit better than StreamCM. This can probably be attributed to the more time that users spent with StreamCW than with StreamCM (Fig. 10), among other reasons. EnhancedLIC was even better than the two streamline-based techniques in displaying centers, because it supports not only 2D continuity but also strong visual intuition or high image contrast in the form of accentuated thin flow streaks. Under uniform density control, not all streamlines around each center are placed (and shown) as closed curves in a StreamCM/StreamCW image. Thus some insufficiently integrated, open streamlines may prevent the user from seeing the closed ellipse(s) around a center.

4.3 Symmetric Pattern Categorization (SPC)

We measured response time and categorization error during SPC sessions. Under the fixed-duration test scheme, the response time is an average value, equal to 30 seconds divided by the number of SPC tasks completed in a session. A categorization error occurs if a wrong symmetry type is selected for a flow image. Figures 11-12 show the number of categorization errors and mean response time, respectively, for each of the seven techniques. In correctly categorizing symmetric patterns, EnhancedLIC and StreamCM were equally effective, marginally preceding BasicLIC. BasicLIC slightly outperformed OrientedLIC. After OrientedLIC were StreamCW and ArrowCM. ArrowCW was the worst for SPC tasks. With regard to time, users responded the slowest with ArrowCW. The other six led to statistically approximate user responses, with an exception that StreamCW required more time than StreamCM, BasicLIC, and EnhancedLIC. As users were engaged in these challenging global+local flow analysis tasks, they spent more time on SPC than on CPR/CPC (Fig. 7 and Fig. 10). Based on the COSS rule, the seven techniques may be sorted by SPC effectiveness in decreasing order as

- EnhancedLIC, StreamCM, BasicLIC, OrientedLIC, StreamCW, ArrowCM, ArrowCW

where EnhancedLIC and StreamCM are equally effective, as are StreamCW and ArrowCM.

Performing an SPC task may involve four steps: (globally) recognize a critical point p; (semi-globally) detect if there is another critical point q that seems to form an x-axis/y-axis/center symmetric pair with p; (locally) classify critical points p and q; (semi-globally) determine if p and q match in the type. Since CPR and CPC are sub-tasks of SPC, our discussions about CPR effectiveness (Section 4.1) and CPC effectiveness (see Section 4.2) apply to the analysis of SPC effectiveness. Special treatment is given below to some issues that we feel made a great impact on the SPC test result. First, the switches between global, semi-global, and local sub-tasks impose memory overhead on the participant. This side effect seemed to degrade the visual intuition of EnhancedLIC as users were comparing the shape of two critical points. Consequently StreamCM caught up with EnhancedLIC in SPC, despite being left behind in CPR and CPC. Second, the visual distraction that the colorwheel introduces, mostly at a global scale (Section 4.1), increases as multiple global/semi-global sub-tasks are executed during SPC. This might partially account for the big gap between ArrowCW and ArrowCM, and for ArrowCM being as effective as StreamCW. Third, although the rainbow color map is based on the velocity magnitude, it may aid in perceiving the flow direction due to some degree of correlation between the flow direction and the velocity magnitude. Its use in ArrowCM, BasicLIC, OrientedLIC, StreamCM, and EnhancedLIC probably contributed to narrowing the gaps between the former three and the latter two since some users might try using color patterns to accelerate sub-tasks 2 and 4 of SPC (though they were not trained or told such). On one hand, the SPC test result may reiterate the importance of color mapping in a flow representation. On the other hand, it also indicates some issues that are usually encountered when designing a sophisticated user study, e.g., the trade-off between a number of techniques (7), resources (time duration and task intensity of each test), and a wide range of issues of interest (color mapping, continuity of flow representation, coverage of recent advances, flow analysis task design, etc.).

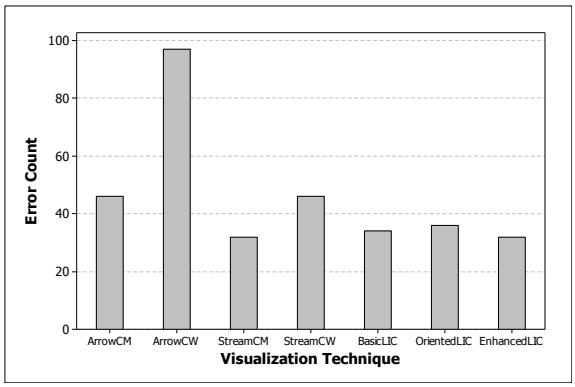


Fig. 11. Number of SPC errors for each visualization technique (N = 323). The error count was different for each visualization technique ($\chi^2(6) = 70.1$, $p < 0.001$). EnhancedLIC and StreamCM were the best to correctly categorize symmetric patterns. Note the considerable difference that color mapping made.

4.4 New Insights and Pragmatic Lessons

Although Laidlaw et al.'s work [4] and ours address 2D flow visualization user studies, ours uses explicit flow synthesis, implicit task design, diverse evaluation perspectives, a hybrid timing strategy, and a refined statistical analysis. The significant difference in the set of visualization techniques prevents a direct comparison in the evaluation results and findings. On the other hand, this difference allows us to compare the two in evaluation perspectives that are largely governed by the selection of visualization techniques. The phrase "evaluation perspectives" refers to the intrinsic visual aspects of a flow repre-

sentation, which, closely related to and helpful for visualization algorithm design and improvement, are investigated via flow analysis tasks in a user study. They may involve color mapping, representation continuity, visual intuition, and image contrast.

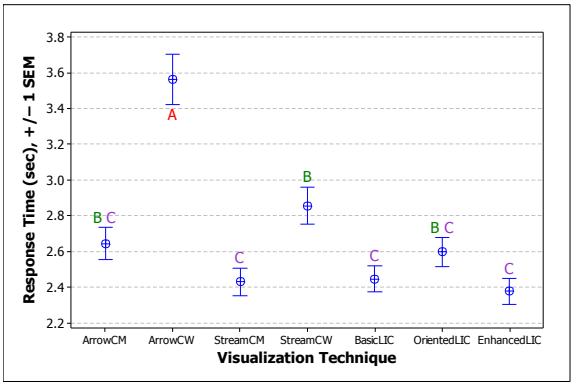


Fig. 12. Mean time (in seconds) to categorize a symmetric flow pattern (N = 4948). There was a main effect of visualization technique on the time to categorize the flow pattern symmetry ($F(6,123.1)$ = 8.74, $p < 0.001$). Means with the same letter are not significantly different at $p \leq 0.05$ (Ryan REGWQ post-hoc homogeneous subset test). ArrowCW caused the slowest user response to an SPC task. ArrowCM was comparable to streamlines and LIC in user response.

Laidlaw et al.'s user study [4] employs three hedgehog techniques (grid-based arrows, randomly distributed arrows, and wedges), two *streamlet* (very short streamline) techniques (grid-based and evenly-spaced), and basic LIC. First, they all utilize gray-scale encoding, without addressing color mapping. Second, the small size of streamlets causes discontinuities along the flow direction and hence the two streamlet techniques may not be used to evaluate 1.5D representation continuity. Third, besides the hedgehog techniques, neither the streamlet techniques nor basic LIC provides good visual intuition for locating and classifying critical points. Fourth, the mere selection of basic LIC (without enhanced LIC) and the lack of color mapping prohibit a study in image contrast. The visualization techniques selected in our user study support diverse evaluation perspectives (Sections 4.1, 4.2, and 4.3) such as the four aforementioned aspects, bringing new insights into 2D flow visualization techniques.

We also learned some pragmatic lessons. Despite informative training sessions, a priori familiarity of the participant with the technique may remain a factor in the evaluation results. Our participants were familiar with arrows and streamlines, though many of them still did not know about LIC. This strengthens the necessity for more user studies and raises concern when evaluating more current techniques [1], [3]. Likewise, a priori familiarity of the participant with specific flow features may be an issue that needs to be taken into account when adding tasks involving complex flow features (e.g., shear layers). This supports the usefulness of synthetic tasks. Although it is useful for synthetic flows to be used in a user study, some real flows (with contextual boundaries) in addition to synthetic ones may be used in training sessions to enhance the participant's understanding of the visualization techniques. Also, care needs to be taken to predict the overall duration of a user study. Statistical data analysis requires a sufficient number of sessions and tasks while fatigue effects need to be avoided. According to some participants, our user study was a little bit long, with the average overall duration being about 90 minutes covering the training, formal test, and breaks.

5 Conclusions and Future Work

We have presented a 2D flow visualization user study that evaluated grid-based variable-size arrows, evenly spaced streamlines, basic LIC, oriented LIC, and enhanced LIC. We chose techniques that reflect some important advances in geometry-based and texture-based flow visualization, addressing 0D, 1.5D and 2D continuity of representation. Our selection, involving the colorwheel and rainbow color maps, allows us to compare seven techniques in visualization effectiveness and explore the impact that each color map has on flow representation.

We employed two new methodologies, i.e., explicit flow synthesis and implicit task design, to make this user study as objective as possible. Explicit flow synthesis uses template-based parameterized specification of critical points to produce many diverse, relatively balanced x-axis/y-axis/z-axis symmetric and even geometrically symmetric but topologically asymmetric flows. Its capability of maintaining approximate topological complexity between flows combats data-related bias. Implicit task design adopts pattern-based flow analysis such as critical point recognition, critical point classification, and symmetric pattern categorization to 'force' the participant to inspect the flow direction both extensively and intensively. In contrast with sample-based task design, this indirect high-level mechanism diminishes task-related bias by avoiding the influence that variance of flow complexity (or an explicitly specified region) may have on visual analysis performance.

To help differentiate one technique from another in visualization effectiveness, we utilized a session-based test strategy that incorporates variable-duration with fixed-duration performance measurement schemes, with the former for relatively quick precision-critical flow analysis and the latter for relatively slow judgment-intensive flow analysis. We present the evaluation results by fixing outliers, in combination with the Ryan REGWQ post-hoc homogeneous subset tests, to derive the findings as accurately as possible. In addition, we discuss the reasons for the findings by combining our understanding of the techniques and the feedback from the participants. Three task-wise correctness-over-speed orderings of the techniques reveal that a texture-based dense representation with crispy thin flow streaks such as enhanced LIC and a geometry-based integral representation with uniform density control such as evenly spaced streamlines are most effective for 2D flow visualization. It is also shown that color mapping plays a very important role in overall flow representation and that in most cases the rainbow color map is better than the colorwheel.

User studies of flow visualization techniques are an emerging topic and there may still be some issues with our work as we try our best to create a convincing user study. This preliminary effort laid a foundation for us to

improve on the selection of visualization techniques, flow data synthesis, and flow analysis task design in future work. It may be useful to compare various 2D streamline placement algorithms surveyed in [28]. The algorithm proposed in [28] allows us to design real tasks for investigating complex flow features such as separatrices and periodic orbits. It is worthwhile to evaluate (direct) feature extraction techniques [12] with dense flow representations [1] as the context. We also plan to adopt surface flow generators [14], [29], [30] and devise new implicit tasks for a surface flow visualization user study. In particular, these flow synthesizers can remove [14], [29] or minimize [30] unexpected critical points such as saddles derived from the interaction among centers and foci. There are several techniques [9], [10] to be evaluated and new issuess to be addressed, such as the choice between fixed viewpoints and arbitrary user navigation, which will influence the perception of surface flows in 3D.

ACKNOWLEDGMENT

This work was supported by the High Performance Computing and Visualization Initiative (HPCVI) funding provided by the DoD High Performance Computing Modernization Program (HPCMP) through the Army Corps of Engineers Engineering Research and Development Center (ERDC) and Jackson State University. The authors would like to thank David Laidlaw for his suggestions on designing the user study and developing the test system, and the anonymous reviewers for the valuable comments.

REFERENCES

[1] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. 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.

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

[3] 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.

[4] D. H. Laidlaw, R. M. Kirby, C. D. Jackson, J. S. Davidson, T. S. Miller, M. D. Silva, W. H., Warren, and M. J. Tarr, "Comparing 2D Vector Field Visualization Methods: A User Study," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 1, pp. 59-70, 2005.

[5] A. Forsberg, J. Chen, and D. Laidlaw, "Comparing 3D Vector Field Visualization Methods: A User Study," *IEEE Trans. Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1219-1226, 2009.

[6] C. Ware, "Toward A Perceptual Theory of Flow Visualization," *IEEE CG&A*, vol. 28, no. 2, pp. 6-11, 2008.

[7] C. Johnson, R. Moorhead, T. Munzner, H. Pfister, P. Rheingans, and T. S. Yoo, "NIH/NSF Visualization Research Challenges," *IEEE Computer Society*, Los Alamitos, CA, ISBN: 0-7695-2733-7, pp. 1-36, 2006.

[8] J. J. van Wijk, "Image Based Flow Visualization," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 745-754, 2002.

[9] J. J. van Wijk, "Image Based Flow Visualization for Curved Surfaces," *Proc. IEEE VIS'03*, pp. 123-130, 2003.

[10] R. S. Laramee, J. J. van Wijk, B. Jobard, and H. Hauser, "ISA and IBFVS: Image Space Based Visualization of Flow on Surfaces," *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 6, pp. 637-648, 2004.

[11] J. Helman and L. Hesselink, "Representation and Display of Vector Field Topology in Fluid Flow Data Sets," *IEEE Computer*, vol. 22, no. 8, pp. 27-36, 1989.

[12] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The State of the Art in Flow Visualization: Feature Extraction and Tracking," *Computer Graphics Forum,* vol. 22, no. 4, pp. 775-792, 2003.

[13] R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post, "Topology-Based Flow Visualization — The State of the Art," in *Visualization and Mathematics* (editors H. Hauser, H. Hagen, and H. Theisel), Springer-Verlag, pp. 1-19, 2007.

[14] E. Zhang, K. Mischaikow, and G. Turk, "Vector Field Design on Surfaces," *ACM Trans. Graphics*, vol. 25, no. 4, pp. 1294-1326, 2006.

[15] K.-L. Ma, B. Cabral, H.-C. Hege, V. Interrante, and D. Stalling, "Texture Synthesis with Line Integral Convolution", *SIGGRAPH'97, Course Notes*, 1997.

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

[17] A. Okada and D. L. Kao, "Enhanced Line Integral Convolution with Flow Feature Detection," *Proc. IS&T / SPIE Electronics Imaging'97*, vol. 3017, pp. 206-217, 1997.

[18] Z. Liu, R. J. Moorhead II, and J. Groner, "An Advanced Evenly Spaced Streamline Placement Algorithm," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 965-972, 2006.

[19] C. Ware, "Information Visualization: Perception for Design," *Morgan Kaufman Publisher*, 2004.

[20] A. Johannsen and R. J. Moorhead II, "AGP: Ocean Model Flow Visualization," *IEEE CG&A*, vol. 15, no. 4, pp. 28-33, 1995.

[21] G. Turk and D. Banks, "Image-Guided Streamline Placement," *Proc. SIGGRAPH'96*, pp. 453-460, 1996.

[22] B. Jobard and W. Lefer, "Creating Evenly Spaced Streamlines of Arbitrary Density," *Proc. Eighth Eurographics Workshop on Visualization in Scientific Computing*, pp. 45-55, 1997.

[23] H. Wainer and D. Thissen, "Graphical Data Analysis," *A Handbook for Data Analysis in the Behavioral Sciences: Statistical Issues* (editors G. Keren and C. Lewis), Lawrence Erlbaum Associates, pp. 391-458, 1993.

[24] M. E., J. Masson and G. R. Loftus, "Using Confidence Intervals for Graphically Based Data Interpolation," *Canadian Experimental Psychology*, vol. 57, no. 3, pp. 203-220, 2003.

[25] V. Barnett and T. Lewis, "Outliers in Statistical Data (3rd Edition)," *John Wiley and Sons*, 1994.

[26] D. C. Howell, "Statistical Methods for Psychology (5th Edition)," *Duxbury Publisher*, 2002.

[27] Z. Liu and R. J. Moorhead II, "Robust Loop Detection for Interactively Placing Evenly Spaced Streamlines," *IEEE Computing in Science and Engineering*, vol. 9, no. 4, pp. 86-91, 2007.

[28] K. Wu, Z. Liu, S. Zhang, and R. J. Moorhead II, "Topology-Aware Evenly-Spaced Streamline Placement," *IEEE Trans. Visualization and Computer Graphics*, vol. 16, no. 5, pp. 791-801, 2010.

[29] J. Palacios and E. Zhang, "Rotational Symmetry Field Design on Surfaces," *ACM Trans. Graphics*, vol. 26, no. 3, article #55, 2007.

[30] N. Ray, B. Vallet, W. C. Li, and B. Levy, "N-Symmetry Direction Field Design," *ACM Trans. Graphics*, vol. 27, no. 2, article #10, 2008.



Zhanping Liu is a research fellow with the School of Medicine at the University of Pennsylvania. Previously he was a research staff member with Kitware, Inc. and Visualization, Analysis, and Imaging Lab at Mississippi State University, and much earlier a post-doctoral associate with the Micro-CT Lab at the University of Iowa. He received the PhD degree in Computer Science (2000) and the B.S. degree in Mathematics (1992) from Peking University and Nankai University, respectively, P. R. China. His research interests lie in computer graphics and scientific visualization, particularly flow (or vector field) visualization involving geometry-based and texture-based methods.



Shangshu Cai received the B.S. and M.S. degrees in Electrical Information Engineering from University of Science and Technology of China in 2000 and 2003, respectively, and the PhD degree in Computer Engineering from Mississippi State University in 2009. He is currently a post-doc at the University of California, Santa Barbara. His research interests include scientific visualization, information visualization, and image processing.



J. Edward Swan II received the PhD degree in computer science, with specializations in computer graphics and human-computer interaction, from Ohio State University in 1997. From 1997 through 2004 he was a Scientist with the Virtual Reality Lab at the Naval Research Laboratory. He moved to Mississippi State University in 2004, where he serves as an Associate Professor of Computer Science and Engineering and an Adjunct Associate Professor of Psychology. He is also a Fellow of the Institute for Imaging and Analytical Technologies. His research centers on perception in virtual environments and visualization.



Robert J. Moorhead II received a BSEE from Geneva College in 1980 and an MSEE and his Ph.D. from North Carolina State University in 1982 and 1985 respectively. He is currently the Billie J Bill Professor of Electrical and Computer Engineering at Mississippi State University, as well as Director of both the Geosystems Research and Northern Gulf Institutes. He was previously a Research Staff Member at the IBM T.J. Watson Research Center. His current research interests include computationally demanding visualization and analysis issues, proving the value of visualization, and showing the relative value of different visualization techniques. Dr. Moorhead has published more than 100 peer-reviewed manuscripts.



Joel P. Martin received his B.S. (2002) and M.S. (2008) degrees in Computer Engineering from Mississippi State University. He is currently employed by Lockheed Martin, Inc. at the U.S. Army Research Laboratory in Aberdeen, MD. His current research interests include large data and remote visualization.

T.J. Jankun-Kelly is an associate professor of computer science and engineering within the Bagley College of Engineering, Mississippi State University. His research is at the intersection of scientific and information visualization; his goal is to make visualization more effective by improving interaction methods and visualization utilization. T.J. has a Ph.D. from the University of California, Davis and a B.S. from Harvey Mudd College.

