Comparing 2D Vector Field Visualization Methods: A User Study

David H. Laidlaw, Robert M. Kirby, Cullen D. Jackson, J. Scott Davidson, Timothy S. Miller, Marco da Silva, William H. Warren, and Michael J. Tarr

Abstract—We present results from a user study that compared six visualization methods for two-dimensional vector data. Users performed three simple but representative tasks using visualizations from each method: 1) locating all critical points in an image, 2) identifying critical point types, and 3) advecting a particle. Visualization methods included two that used different spatial distributions of short arrow icons, two that used different distributions of integral curves, one that used wedges located to suggest flow lines, and line-integral convolution (LIC). Results show different strengths and weaknesses for each method. We found that users performed these tasks better with methods that: 1) showed the sign of vectors within the vector field, 2) visually represented integral curves, and 3) visually represented the locations of critical points. Expert user performance was not statistically different from nonexpert user performance. We used several methods to analyze the data including omnibus analysis of variance, pairwise *t*-tests, and graphical analysis using inferential confidence intervals. We concluded that using the inferential confidence intervals for displaying the overall pattern of results for each task measure and for performing subsequent pairwise comparisons of the condition means was the best method for analyzing the data in this study. These results provide quantitative support for some of the anecdotal evidence concerning visualization methods. The tasks and testing framework also provide a basis for comparing other visualization methods, for creating more effective methods and for defining additional tasks to further understand the tradeoffs among the methods. In the future, we also envision extending this work to more ambitious comparisons, such as evaluating two-dimensional vectors on two-dimensional surfaces embedded in three-dimensional space and defining analogous tasks for three-dimensional visualization methods.

Index Terms—User study, vector visualization, fluid flow visualization.

1 Introduction

One of the goals of scientific visualization is to display measurements of physical quantities so the underlying physical phenomena can be interpreted accurately, quickly, and without bias. Great care is taken in choosing where such measurements will be made so that inferences about the underlying phenomena will be correct. How important is it to craft visualizations analogously, carefully placing arrows, curves, or other visual icons that display the data? What are the best ways to craft visualizations?

Many people have addressed, with qualitative or anecdotal advice, how best to design visualizations [1], [2], [3]. For example, Ware suggests that vectors placed on a regular grid are less effective than vectors placed in a streamline-like (or integral curve) fashion. Analogous quantitative studies of visualization methods are still very limited [4], [5], [6], [7], and none address 2D vector visualization methods. Albeit limited, such quantitative studies help to form a basis upon which rule-of-thumb construction measures for vector visualizations can be postulated.

An earlier version of the study presented here included only nonexpert users, did not comparatively include LIC in the analysis, did not include an analysis of user performance as a function of flow speed at the user-chosen critical point locations, included arbitrarily difficult tasks involving critical points close to the boundary of the visualization and reported a pessimistic analysis of counting accuracy [8]; we address those limitations in this paper with new analyses that go beyond the initial analyses accomplished for that version.

Our quantitative study of these questions began with a (naive) hypothesis of the form "When visualizing two-dimensional vector fields, arrows spatially distributed using method X are more effective than arrows spatially distributed using method Y." We proposed to test the hypothesis with a user study. The first hurdle which stymied our progress was an understanding of formulating and executing a task-based user study. How does one define "more effective?" Can "more effective" be established in a broad-brush fashion, or is it possible to construct tasks in which for task A method X is more effective than method Y (in some metric), while for task B method Y is more effective than method X? After much deliberation, we decided to define "more effective" via the performance of users on a set of three tasks derived from examination of common comparison metrics used in

- *D.H. Laidlaw, C.D. Jackson, J.S. Davidson, T.S. Miller, and M. da Silva are with the Computer Science Department, Brown University, Providence, RI 02912. E-mail: {dhl, cj, jsdavids, tsm, mds}@cs.brown.edu.*
- *R.M. Kirby is with the Scientific Computing and Imaging Institute and School of Computing, University of Utah, Salt Lake City, UT 84112. E-mail: kirby@cs.utah.edu.*
- *W.H. Warren and M.J. Tarr are with the Cognitive and Linguistic Sciences Department, Brown University, Providence, RI 02912. E-mail: {Michael_Tarr, William_Warren}@brown.edu.*

Manuscript received 3 Sept., 2003; revised 18 Feb. 2004; accepted 19 Feb. 2004. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-0088-0903.

flow visualization and based upon domain expert input as to what are representative tasks within the area of fluid mechanics. These three tasks and the rationale are thoroughly described in Section 3. If users could perform the tasks more accurately and quickly using one of the methods, we would consider that method more effective with respect to the task. "X" and "Y" were initially the first two methods in the list below, but as we designed the experiment, we realized that broader coverage of the existing methods would be more valuable. We converged on the following six visualization methods:

1. GRID: icons on a regular grid,
2. JIT: icons on a jittered grid [9],
3. LIT: icons using one layer of a visualization method that borrows concepts from oil painting [10],
4. LIC: line-integral convolution [11],
5. OSTR: image-guided streamlines (integral curves) [12], and
6. GSTR: streamlines seeded on a regular grid [12].

Henceforth, we will refer to each visualization method by its abbreviated name.

2 Vector Fields and Visualization Methods

2.1 Vector Field Data Set Generation

To accomplish the user study, we required a controlled set of stimuli. We first generated approximately 500 two-dimensional vector field data sets from which images could be generated. By first generating a database of fields, we could then create for any particular vector field six different visualizations, one for each visualization method to be used as stimuli.

We used Matlab [13] to generate the vector field data sets. Each data field was represented by a regular grid of 700 by 700 vectors and was generated in the following manner. For each data set, nine random locations on the interval $[0,1] \times [0,1]$ were chosen. This was accomplished for each of nine locations by randomly choosing, with uniform distribution, a position on the *x*-axis and then randomly choosing, with uniform distribution, a position on the *y*-axis. At each random location, a vector was generated such that both components of each random vector were chosen from a uniform random distribution between -1.0 and 1.0. The x and y components of these nine vectors, along with a regular grid of 700 by 700 uniformly spaced points, were input to the Matlab function `griddata` using the "v4" option (for Matlab's spatial interpolating function), which, in turn, provided x and y vector components for the interpolated vector field at each of the 700 by 700 grid points.

To calculate the user accuracy on the critical-point tasks, we needed to know the correct locations and types of all critical points within these vector field data sets. The critical points were located in each vector field using a two-dimensional Newton-Raphson (root finding) method. In the Newton-Raphson method, second-order finite differences are used to form the gradient and Jacobian information required. We used 150 random initial positions for each field and iterated the Newton-Raphson solver until a critical point (root) of the vector field was found. Once a critical point was located and verified to be a critical point, Matlab routines based on second-order finite differences formed the local Jacobian of the field at the critical point and determined the eigenvalues of the Jacobian, which determine the type of a critical point. This method was verified against the TOPO module of the FAST visualization environment [14] for several of the fields, and showed no errors. Data fields were discarded if they contained fewer than one or more than four critical points.

2.2 Visualization Methods

Six visualizations were generated for each vector field, one for each visualization method. The visualizations for one vector field are shown in Fig. 1. Both GRID and JIT were generated using standard Matlab plotting routines. LIC [11] and LIT [10] were implemented by the authors from published descriptions. OSTR and GSTR were actualized using code from Turk and Banks [12].

Each of the visualization methods has parameters that influence the images created (e.g., the path integration length in LIC, or the grid spacing in GRID). For each visualization method, we created three test images over five values in a range for each parameter. Five of the authors independently and subjectively estimated which value of a parameter would be best for performing each of the tasks. We then viewed them as a group and came to a consensus value for each parameter based on the tasks that we were planning. We were generally in accord on the best parameter value for a given task but that setting sometimes differed across tasks. We tried to choose a compromise value that would work as well as possible for all three tasks. For example, this process led to a 29×29 lattice for GRID and a 35×35 lattice for JIT; tighter spacing in JIT worked better for the collection of tasks. The following paragraphs describe the parameter values we chose.

For GRID, a uniformly-spaced lattice of 29×29 points was used to span $[-1,1] \times [-1,1]$. To find the x and y values of the vector at each of the given points in the lattice, Matlab's `interp` routine with "spline" setting was used to interpolate down from the 700×700 point data set to the 29×29 point data set. The vectors were created by giving the x, y, v_x, v_y arrays to the Matlab routine `quiver`, which graphically displays vectors as arrow icons. The automatic scaling provided by `quiver` was used; no special parameters were passed to `quiver`.

For JIT, a uniformly-spaced lattice of 35×35 points was used to span $[-1,1] \times [-1,1]$. For each point (x,y), an offset was computed in both the x and y directions. The offset was uniformly distributed in $[-\frac{\delta}{2}, \frac{\delta}{2}] \times [-\frac{\delta}{2}, \frac{\delta}{2}]$, where δ denotes the spacing between uniformly-spaced grid points. Once a jittered grid was created, both the Matlab `interp` and `quiver` functions were used to interpolate and graphically represent the vectors, as in the uniform grid (GRID) case.

For LIT, a triangle-shaped wedge, with a base one-quarter its length, represented the flow at points in the field. The area of each wedge was proportional to the speed at its center, and the wedges were placed using a uniform random distribution such that they would overlap at most 40 percent along their long direction and would maintain clear space between wedges of at least 70 percent of their width. Wedges that would not have satisfied this spacing were not kept. Strokes

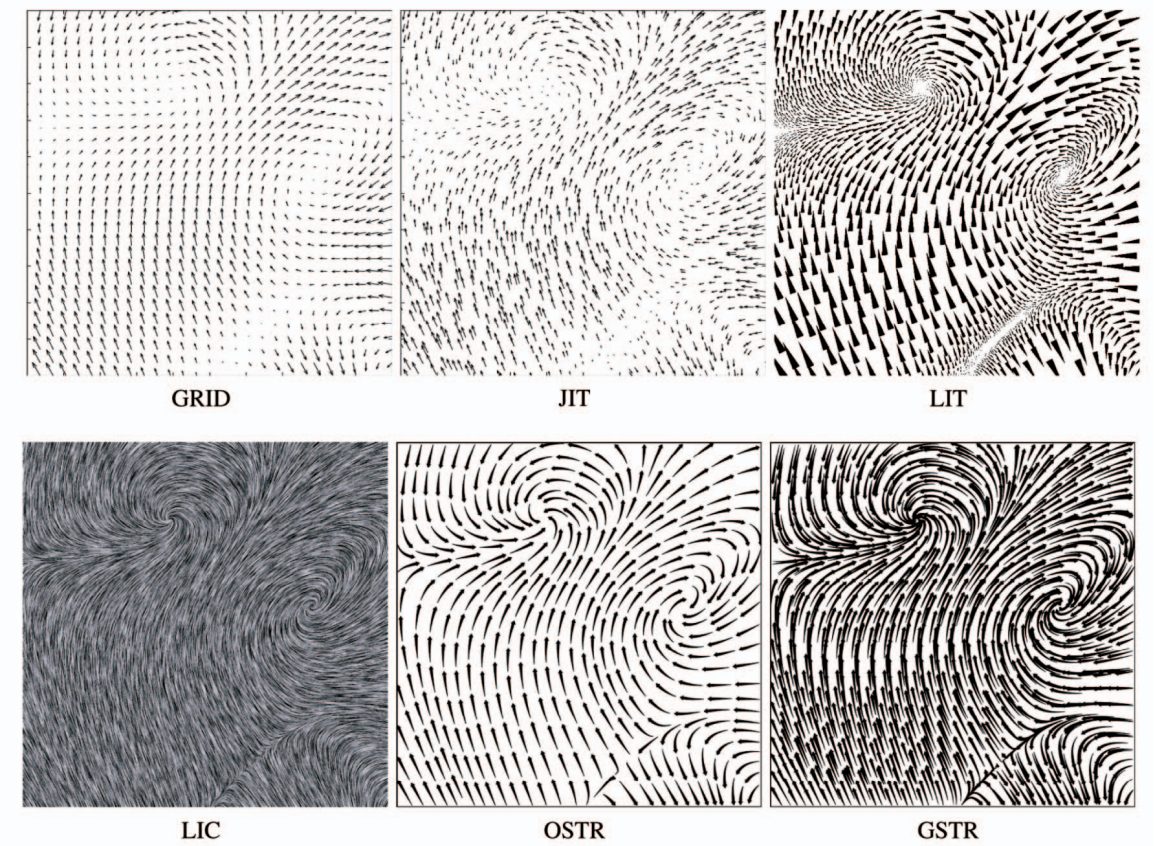

Fig. 1. One of the approximately 500 vector fields visualized with each of the six visualization methods.

were placed until 250 consecutive attempts failed the spacing criteria. The overall size of the wedges was scaled so that there would be about 2,000 strokes in each image.

For LIC, we used a box-shaped convolution kernel of width 20 pixels. The convolution was performed on a noise image where each pixel value was set to a uniform random value in the interval $[0, 1]$. To correct for loss of contrast due to the convolution, we applied an intensity mapping that took intensity I to $I^{(4/(I+1)^5)}$.

For OSTR and GSTR, the code from [12], version 0.5, was modified to allow batch running without a graphical display and to have the optimization process stop after 60 seconds, without requiring manual intervention. OSTR was invoked with `opt 0.017` given to the `stplace` program (the "opt 0.017" parameter invokes optimal streamline placement using the algorithm from reference [12] with a separation choice of 0.017), while GSTR was invoked with `square 23 .2` (streamlines 20 percent of the image width each centered on a square grid of 23 points in each direction), and both were plotted with "fancy arrows." All other options to OSTR and GSTR were left as the defaults.

3 Two-Dimensional Vector Tasks

The tasks we used to evaluate the effectiveness of visualization methods needed to be representative of typical interactions that users perform with visualizations, simple enough that users could perform them enough times for us to calculate meaningful statistics, and able to provide an objective measure of accuracy.

We chose fluid mechanics as our "representative" scientific field because it frequently utilizes vector visualizations as a means of studying physical phenomena. We searched the literature and interviewed fluid mechanics researchers to identify good representative tasks. Two of the tasks, locating critical points and identifying their types, were derived from motivations behind the development of many of the visualization methods that we tested. Critical points are the salient features of a flow pattern; given a distribution of such points and their types, much of the remaining geometry and topology of a flow field can be deduced, since there is only a limited number of ways to join the streamlines. Beyond their importance for the interpretation of vector fields, these tasks are testable: we can measure how accurately a user determines the number, placement, and type of a collection of critical points in a given image.

Fig. 2 shows an example stimulus for locating all the critical points in a vector field. The GSTR method is used in this example. Users indicated the location of each critical point with the mouse and pressed the "Enter" key (or clicked on the "Next Image" button) when finished. Users were not allowed to delete or move their chosen points because editing operations tend to significantly increase the variability of response times, making statistical comparisons more difficult. We realized that this limitation on the user's interactions might reduce accuracy but we felt that the benefit of more precise timing was an appropriate

Fig. 2. The experimental setup for the critical point location task. The user was instructed to use the mouse to indicate the locations of all the critical points in each given vector field. The GSTR method is shown but each user viewed all six methods during the course of the experiment.

tradeoff. The locations of all chosen points and the time to choose them were recorded.

Fig. 3 shows an example stimulus for identifying the type of a critical point in a vector field. A preimage with a red dot indicating the location of the critical point to identify appeared for 500 ms before the visualization. The user then selected the type of critical point from the five choices at the bottom of the display using the mouse: attracting focus, repelling focus, attracting node, repelling node, and saddle. The critical point type selected and the response time were both recorded.

In addition to these critical-point tasks, we identified a third task that is different in character from the other tasks, yet important in interpreting two-dimensional vector fields—an advection task. This task is motivated by an implicit criterion sometimes mentioned when qualitatively examining visualization methods: The ability of a method to show the flow direction globally. In this task, the user is presented with a dot inside a circle. The user must identify where the dot will intersect the circle as it is carried by the flow.

Fig. 4 shows an example stimulus for performing the advection task with the LIT method. The user chose the point on the circle where a particle, advected from the center dot, would intersect the circle. The user chose the intersection point using the mouse and then pressed the "Enter" key on the keyboard or the "Next Image" button on the screen to continue to the next stimulus. The coordinates of the chosen point and the response time were both recorded. A small red icon in the lower left corner indicated the signed direction of the vector field at the location of the icon. This is needed for LIC, which does not capture the sign of the vector field; it is included in all of the methods to avoid biasing results.

In summary, the three tasks are:

- choosing the number and location of all critical points in an image,
- identifying the type of a critical point at a specified location, and
- predicting where a particle starting at a specified point will advect.

The tasks that we have chosen are testable and, we believe, representative of many of the real uses of these visualizations. As such, they are potentially predictive of the performance of real users in using these kinds of visualizations. Of course, these three tasks do not encompass all possible tasks for which a fluids scientist would use vector visualization. For example, combinations or modified versions of these tasks may be more or less difficult than straightforward generalizations would predict. However, performance on these tasks seems reasonably likely to generalize to performance on other similar or aggregate tasks.

4 Experimental Details

4.1 Timing and Training

Fig. 5 shows the timing of the study. Users first saw a text display for general training describing the goals of the experiment and the three tasks in general terms. Three parts of the experiment followed these instructions, one part for

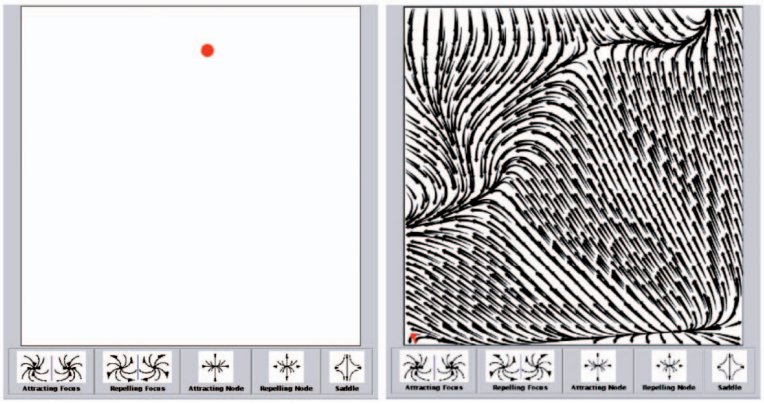

Fig. 3. The experimental setup for the critical point type identification task. A red dot indicating the critical point to identify appeared in a blank frame (the first panel) for 500 ms before the visualization was displayed (the second panel). The user chose the type for that critical point from the list of icons at the bottom by clicking on the corresponding icon. The GSTR method is shown but each user viewed all six visualization methods during the course of the experiment.

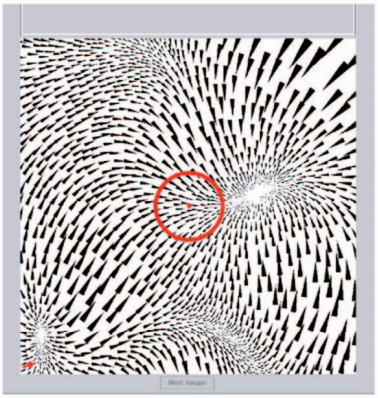

Fig. 4. The experimental setup for the advection task. The user chose the point on the circle where a particle would end up if advected from the center of the circle. The LIT method is shown but each user viewed all six visualization methods during the course of the experiment.

each task. Within each part, an initial text display described the task in more detail. For each task, the six visualization methods were tested each in a separate block of trials. The task instructions were followed by eight untimed training trials; in a pilot study, two users were found to converge to reasonable accuracy after eight example stimuli. For each of these untimed cases, the correct answer was provided after the user completed the task so that users were consistently trained before the timed tasks. After the training period, the user performed 20 timed instances of the task. Users performed this training/testing sequence for each visualization method before moving on to the next task.

A Java program written specifically for this experiment presented the stimuli and recorded the data. The program pre-loaded all images at the beginning of a block so that timing would be consistent for each stimulus. A several-second pause before each block, however, did cause some small problems that are discussed later.

To avoid biasing the results, the ordering of tasks and of visualization methods within the tasks were each counterbalanced with a replicated randomized Latin square design [15]. For the testing (timed and recorded) phase of each task, 120 images were generated; each block of 20 images within that 120 was assigned to a visualization type per user, counterbalanced with a randomized Latin square design.

4.2 Participant Pool

Two cohorts of participants were tested: "nonexperts" and "experts." We distinguished between expert and nonexpert users because we hypothesized that experts might perform with a bias toward tools similar to those they already use. For nonexperts, we wanted participants who might use such tools in the future for their work but who had not yet started to do so. Nonexperts were undergraduate science majors that had studied applied math but had not studied fluid mechanics. Experts were faculty or graduate students studying fluid dynamics.

Data for 12 nonexpert participants and five expert participants were successfully acquired and are reported here. Users were compensated for their participation.

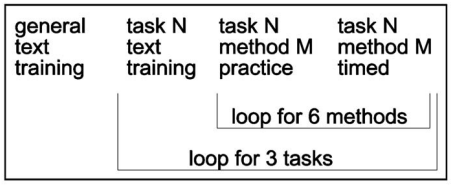

Fig. 5. Ordering of tasks in the experiment.

5 Results and Discussion

A great debate rages concerning the use of standard null hypothesis significance testing techniques for analyzing behavioral data [16], [17]. Some practitioners advocate employing these techniques and claim that researchers misunderstand how to use them correctly [18], [19], [20]. Others suggest that observing the means and standard errors across conditions is sufficient for understanding the pattern of results and the effect of each condition on the dependent measures [21], [22], [23]. We have chosen to use a graphical technique that allows us to make statistical inferences from the confidence intervals displayed around each condition mean. This technique utilizes inferential confidence intervals [20]. With these confidence intervals, the figures indicate a statistically significant difference to the $p = 0.05$ level between any two means if the intervals between the two means do not overlap.

The current study uses a mixed-model or cross-plot design (also called a cross-sectional design in biomedical research). This type of experimental design contains both between-subjects and within-subjects factors. In this design, users participate in only one of the independent levels of the between-subjects factor while participating in all of the levels of the within-subjects factor. In this study, the one between-subjects factor is expertise (expert or nonexpert), and the one within-subjects factor is visualization method (the six methods previously described).

5.1 Expertise

One of the primary interests of this study was to determine if differences existed between expert fluid dynamics researchers and nonexperts in their abilities to perform the three tasks with the six visualization methods. We performed an analysis of variance between the two groups and found no statistically significant differences between the groups for any of the three tasks. Fig. 6 shows a comparison of the visualization methods for the critical point identification task. Results displayed are the means and 95 percent inferential confidence intervals for mean percent error for critical point identification and the associated response time measure. We chose to show this graph as representative of the comparison between the two groups since the mean error for critical point type identification resulted in the largest difference amongst the three tasks ($F(1, 15) = 2.297, p = 0.150$). However, as the figure shows, this difference still was not statistically significant as the confidence intervals between the groups overlap in both graphs.

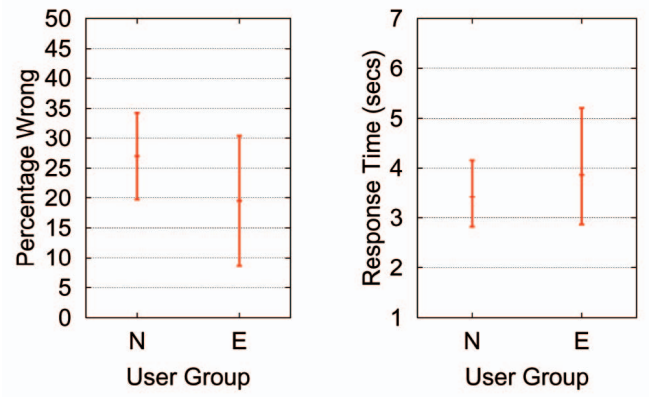

Fig. 6. These graphs show the results of the between-subjects analysis of expertise for the critical point identification task. The first graph shows the mean percentage of incorrect identifications, while the second graph shows the mean response time measure. The error bars shown are 95 percent inferential confidence intervals based on separate mean-squared error terms for each group to compensate for the lack of homogeneity of variance and unequal sample sizes; nonoverlapping bars indicate a statistically significant difference between those means. Both graphs show no statistically significant difference between the two groups. Differences between groups are also not statistically significant for all other measures, with $p > 0.150$.

The analyses of the interactions between the groups and the visualization methods also resulted in no statistically significant differences for all of the tasks. This result implies that both groups exhibited similar response patterns across the visualization methods for all of the tasks. Because we did not observe significant differences between the groups, and found no significant group by visualization method interactions, the remainder of our analyses (figures, statistics, and z-scores) represent the main effect of visualization type.

5.2 Visualization Method

There are several methods for analyzing within-subjects data. We employed several of these techniques, including using analysis of variance (ANOVA) to test the omnibus hypothesis that all methods were not different, followed by pairwise t-tests to directly compare the means of each of the visualization methods to the others, and graphical techniques for displaying the means and confidence intervals for each method. We also utilized several techniques for controlling the Type I error (the probability of rejecting a true null hypothesis) across each task measure. Also, when within-subjects data violate the sphericity assumption, i.e., differences between every pair of condition levels must have the same population variance, the use of the pooled mean-squared error term in any analysis is suspect [15], [24].

For analyzing the effect of the visualization methods on each measure, we used an approach to control for the violation of the sphericity assumption by calculating a standard error estimator based on normalized interaction variances for each condition [24]. Setting the experiment-wise probability of committing a Type I error at 0.05 ($p = 0.05$), and further controlling for multiple comparisons between the condition means ($p = 0.0033$ for each unique pairwise comparison), we obtained 95 percent inferential confidence intervals based on separate mean-squared error terms for each condition. These intervals allow us to graphically determine the statistical differences between the means of the visualization methods; nonoverlapping confidence intervals between two means indicate a statistically significant difference between those two means.

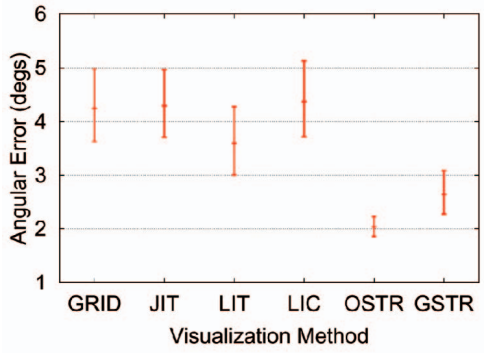

Fig. 7. Mean absolute angular error for the advection task. The error bars shown are 95 percent inferential confidence intervals; nonoverlapping bars indicate a statistically significant difference between those means. The graph indicates that user performance for the advection task was better with OSTR than with any of the other visualization methods, including GSTR. Performance using GSTR was also better than LIC and the two arrow-based methods (GRID and JIT) but not statistically different from LIT.

Figs. 7, 8, 9, 10, 11, 12, 13, and 14 graph the results of the data analysis across the visualization methods. They are organized so that higher values on the vertical axes generally indicate greater error or slower performance (i.e., are worse). The horizontal axis shows the six visualization methods. Mean values are shown with error bars that are 95 percent inferential confidence intervals [20].

Some trials were dropped from the data collected due to timing issues with the software, confounds with the flow field data set, or timing issues with user responses. This is detailed below in the analysis associated with each task.

5.3 Advecting a Particle

One trial was dropped from the data analysis because its associated response time was less than 1,500 ms and probably represented a user responding before the test stimulus was completely loaded. An additional 11 trials were dropped from the analysis for LIC because no possible intersection point existed in the opposite flow direction due

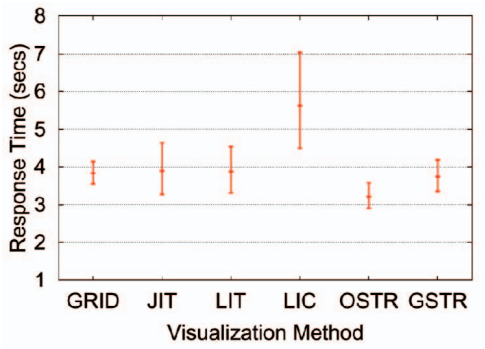

Fig. 8. Mean time to perform the advection task in seconds. The error bars shown are 95 percent inferential confidence intervals; nonoverlapping bars indicate a statistically significant difference between those means. The graph indicates that users took significantly longer to perform the advection task using LIC than using OSTR, GSTR, and GRID.

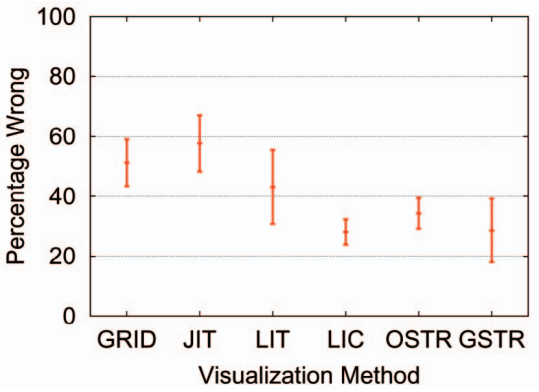

Fig. 9. Percentage of trials in which users incorrectly identified the number of critical points in a stimulus image. The error bars shown are 95 percent inferential confidence intervals; nonoverlapping bars indicate a statistically significant difference between those means. Users committed more errors counting the number of critical points in each image using the GRID and JIT methods than using the LIC, OSTR, or GSTR methods. Mean performance using LIT fell in between these two groups of methods, although the error rate was not statistically different from either group.

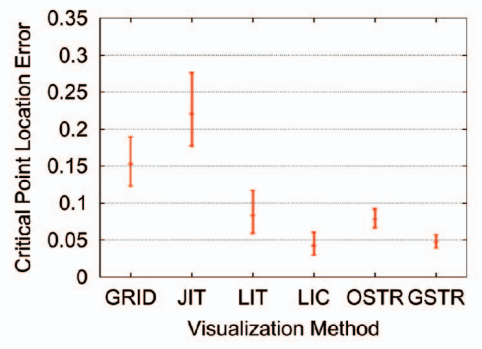

Fig. 10. Mean error magnitude for locating critical points. Error magnitude is the distance between a user-chosen point and the nearest critical point, expressed as a fraction of half of the visualization size. User-chosen points were matched with the nearest real critical point. The error bars shown are 95 percent inferential confidence intervals; nonoverlapping bars indicate a statistically significant difference between those means. Users were worst at locating critical points using the GRID and JIT methods and best using the GSTR and LIC methods (although it should be noted that LIC is not statistically different from LIT). User performance using LIT and OSTR was similar and fell in between the poor performance of GRID and JIT and the good performance of LIC and GSTR.

to a critical point near the calculated point of intersection. A detailed discussion of this analysis follows.

For the advection task, error was measured as the absolute angle error between the user-chosen intersection point and the correct intersection point. In order to normalize the distribution of scores against an observed floor effect, this data, and the associated response time data, were log-transformed before analysis. Also, to compensate for the lack of any cue to flow direction in the LIC method, we calculated the absolute angle error for this method differently from the others. For LIC, we calculated the minimum angle error as the minimum between the absolute angle error measured from the correct intersection point and the intersection point for flow in the opposite direction. Eleven trials were dropped from this analysis because a critical point prevented the calculation of the correct intersection point in the opposite flow direction. The absolute angle error from the opposite flow direction intersection point was used for about 36 percent of the total trials for the LIC method across all users for this task.

Mean angular error results for the advection task are shown in Fig. 7. This figure shows that advection error was greatest with the "icon" methods and LIC. Users exhibited the least error for the task using OSTR. The graph also shows that, while GSTR forced fewer errors than GRID, JIT and LIC, performance using GSTR was not significantly different from performance using LIT.

We conjecture that the LIC method suffered because the images do not display the sign of the vector field. The single directionality icon in the corner of the stimulus is too difficult to propagate across the image to correct for this, as it was intended. Task accuracy is better for OSTR than for the other methods. This may be because the uniform distribution of integral curves throughout the field offers a single integral curve to follow for many advection cases. Most of the other methods require chaining together many icons to perform advection.

Task performance times are shown in Fig. 8. As previously stated, statistics for this measure were also calculated on the log of the response time to normalize against an observed floor effect. Performance with the LIC method was slowest compared to all but JIT and LIT. This slow performance was most likely due to directional ambiguities inherent in the method. In general, the other methods performed similarly.

For the advection task, OSTR is more accurate than the other methods. For response time, most of the methods elicited response times that were not statistically different from each other, with the exception of the LIC method, which was significantly slower than most of the other methods.

5.4 Locating Critical Points

Of 2,040 trials, 114 (5.5 percent) were dropped from the analysis; in each case user response time was less than 2,000 ms, total time recorded for the trial was less than the individual times recorded for locating each critical point, or the user did not make a response.

During data analysis, we realized that we did not control well for critical points near the borders of the images (either inside or outside). We feared that this oversight biased users in choosing critical point locations. We tried several methods to compensate for this bias during data analysis. Several of the methods involved removing individual critical points that were near the image borders, removing user-chosen critical points near the image borders, or performing both data manipulations in an analysis run. We also tried dropping all trials associated with data sets that two authors determined contained ambiguous information near the image borders. None of these data manipulation techniques yielded results significantly different from the original analysis that included all trials and critical points. The following analyses and figures contain all user data not previously excluded for timing issues, as described in the preceding paragraph.

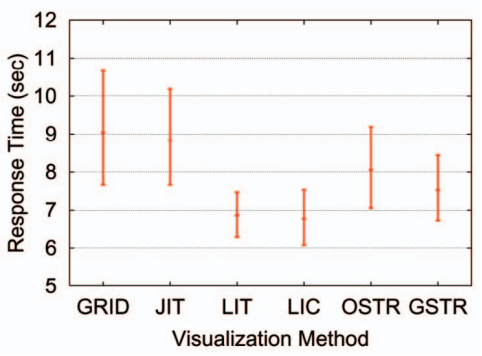

Fig. 11. Mean time to locate critical points. This measure was computed over all trials. The error bars shown are 95 percent inferential confidence intervals; nonoverlapping bars indicate a statistically significant difference between those means. These results show that while users took on average one second longer to perform the critical point identification task using GRID and JIT than using OSTR and GSTR, a statistically significant result was not obtained. However, users did take significantly longer using GRID and JIT than using LIT and LIC.

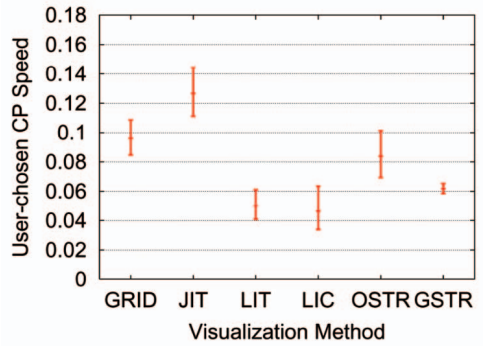

Fig. 12. Mean flow speed at user-chosen critical point locations. This measure was computed over all trials and the log of the speed was used to normalize the distribution prior to analysis. The error bars shown are 95 percent inferential confidence intervals; nonoverlapping bars indicate a statistically significant difference between those means. The results show that the flow speed at user-chosen points was fastest using JIT and slightly slower using GRID and OSTR. Flow speed at the user-chosen points was slowest using LIT, LIC and GSTR. These results indicate that LIT, LIC and GSTR represented the lower speeds around the actual critical points better than the other methods.

Fig. 9 shows the percentage of trials in which users incorrectly identified the number of critical points in the given stimulus image. GRID and JIT are generally least accurate by this measure, although they are not significantly different from the LIT method. LIC, OSTR, and GSTR do similarly well, with users incorrectly locating the critical points in about 30-35 percent of the trials (none of these methods are statistically different from performance with LIT).

A second error measure for this task was the distance from the user-chosen critical points to the actual critical points (see Fig. 10). Statistics for this distance were calculated on the log of the distance as a normalizing transform. Statistics were calculated for all user-chosen critical points such that users' points were matched with the nearest real critical point. A least-squares fit was used to find the closest critical points in all cases.

As demonstrated in Fig. 10, user performance was least accurate for GRID and JIT and most accurate with LIC and GSTR, although the performance difference between LIC and LIT is not statistically significant. Users also showed similar performance for locating critical points using the LIT and OSTR methods and this performance fell between the poor performance of GRID and JIT and the good performance of LIC and GSTR.

Fig. 11 shows performance times for the six visualization methods; statistics were calculated on the log of the time due to the observed floor effect and were taken over all images. While the results with LIC and LIT were not significantly different from the results with OSTR and GSTR, both methods elicited relatively fast user performance. It should be noted that although users spent more time considering the images for GRID and JIT compared to LIT and LIC, users still performed rather poorly in actually locating the critical points using these methods compared to LIT and LIC.

We also performed an analysis on the flow speed at the user-chosen points. The flow speed was calculated by interpolating the velocity based on the user-chosen positions in the given flow field. We believed visualization methods that represented velocity might show slightly better user performance for placing critical points; such methods would represent the decrease in flow speed around the critical points (where flow speed is zero) and allow users to select them more accurately. Fig. 12 shows that LIT, which displays flow velocity most clearly of all the methods, exhibits good user performance. Surprisingly, LIC and GSTR, which do not explicitly display flow velocity, elicited user performance that was not statistically different from user performance using LIT. The results also show that JIT and GRID, which directly display flow velocity, did not elicit good user performance compared to LIT, LIC, and GSTR. These results are also consistent with the results of the analysis based on the distance of user-chosen critical points from the actual points.

Overall for this task, GRID and JIT elicit poor user performance for most of the task measures while the other methods generally exhibit fast and accurate performance.

5.5 Identifying Critical Point Type

Only one trial was dropped from the data analysis for the critical point type identification task. This trial had a user response time of zero, possibly due to the user clicking multiple times at the end of the practice trials (for LIC) and this input propagating to the first test trial. The task response times were log-transformed to correct for an observed floor effect.

The percentage of critical points identified incorrectly by users for each visualization method is shown in Fig. 13. The results show a statistically significant difference in user response times using GSTR compared to LIC and JIT. No other statistically significant differences are apparent in the results.

Fig. 14 shows user response times for each of the visualization methods on the critical point identification task. The results indicate a decreasing pattern of performance, which translates into an overall improvement in user response time, from the "icon" methods to the

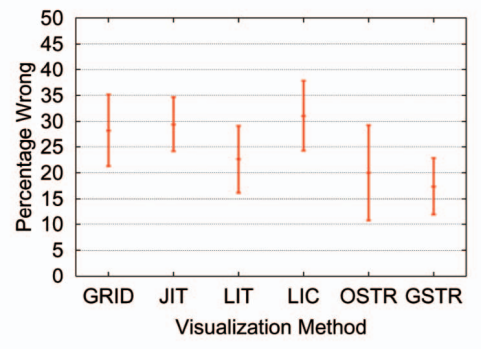

Fig. 13. Percentage of trials in which users misidentified the type of the given critical point. The error bars shown are 95 percent inferential confidence intervals; nonoverlapping bars indicate a statistically significant difference between those means. While the results show a statistically significant difference in performance comparing GSTR to LIC and JIT, no other statistical differences exist among the visualization methods.

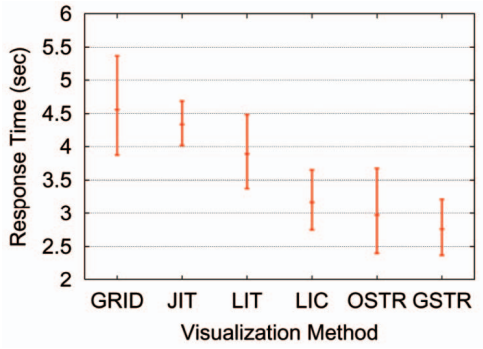

Fig. 14. Mean time to identify critical point type. The error bars shown are 95 percent inferential confidence intervals; nonoverlapping bars indicate a statistically significant difference between those means. The results indicate that users were slower to respond using GRID and JIT than using LIC, OSTR and GSTR. Users also responded faster using GSTR than using LIT.

streamline methods. The results also indicate that user response times using the streamline methods (LIC, OSTR, and GSTR) were significantly faster than using the arrow-based methods (GRID and JIT). GSTR also elicited faster performance than LIT.

For this task, while user critical point identification error was similar across the six visualization methods, user response times were significantly faster for the streamline methods (LIC, OSTR, and GSTR) than for the arrow-based methods (GRID and JIT).

5.6 Analysis Details

Statistics were computed using all of the data (after any transformation such as the logarithm or dropping of trials as previously discussed) of a given user for a given visualization type and task. As discussed above, we chose to analyze the data using a graphical technique that allows us to draw statistical inferences from the confidence intervals displayed around each condition mean. This technique utilizes inferential confidence intervals [20]. The inferential confidence intervals displayed in each figure indicate a statistically significant difference to the $p = 0.05$ level between any two means if the intervals between the two means do not overlap.

For the critical point location task, the number of critical points ranged from one to four with a median value of three and mean of 2.683. For the advection task angular error, Batschelet [25] states that circular statistics are unnecessary for angular differences if the sign of the angular difference does not matter, and therefore standard linear statistics were used on the absolute value of the angular error.

5.7 Normalizing Visualization Methods

We attempted to normalize the visualization methods by setting their parameters to optimal values for our tasks. However, the normalization might have been done more rigorously. With a different tuning of the methods, results might have been different. We did attempt to balance the parameters to perform as well on all of the tasks as possible. A more formal study for each method could have measured performance of users with each method using different parameter settings, allowing us to choose the parameters more objectively.

We considered an alternative normalization as well: creating images that had a comparable "density." However, we were not able to define density for all methods and also found that the optimal parameter settings for different methods produced images with densities that were quite different. As a simple example, the GRID and the JIT methods were very similar, and yet the optimal number of icons differed by 45 percent. Given the difficulties in specifying this approach, we opted for the optimal parameter settings previously described.

5.8 Data

Our randomly constructed two-dimensional vector data sets were not drawn from experimental or computational fluid flow examples. However, they are two-dimensional vector fields, well sampled, with good continuity, and with moderate numbers of critical points. They also contain the different types of critical points in the ratios one would expect for fluid flow data. Fluid researchers found their appearance representative. While an alternative construction or source of data might be interesting to test, particularly if it was representative of incompressible flow or some other constrained type of vector field, we felt that the construction we used was a reasonable compromise.

5.9 Tasks Involving Speed and Other Quantities

The three tasks that we chose did not involve analyzing speed (except loosely, in the case of critical points) or other data values. We chose not to extend the task set, because experiment time would have been prohibitively long and the other tasks were more important.

6 Conclusions

In Fig. 15, we present z-scores calculated for each visualization method for each task. For each task, the z-scores represent the direction and distance of the performance for each visualization method from the mean performance for that task in units of standard deviation. We can use these normalized measures to compare the pattern of performance for each method across tasks. Scores above zero indicate more

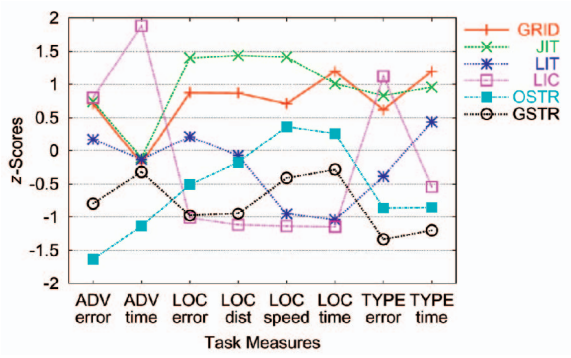

Fig. 15. *z*-scores for each visualization method calculated for each task measure. The *z*-scores represent the difference between the mean performance for a method and the mean performance for all methods for each task. This difference is divided by the standard deviation of the mean performance for each task to provide a measure with units of standard deviation. This transformation allows comparisons between the relative patterns of performance of each method across the task measures. The categories along the *x*-axis represent the task measures: ADV (advection) error and time; LOC (critical point location) counting error, distance error, flow speed error, and time; and TYPE (critical point type identification) error and time. Scores above zero indicate more error or slower performance on the task, while scores below zero indicate less error or faster performance.

error or slower performance on the task, while scores below zero indicate less error or faster performance. From Fig. 15, combined with the previously discussed results, we can infer many things; we now present some of our observations.

Our original motivating hypothesis, that GRID would perform better than JIT for most tasks, was not validated—user performance was statistically different for these two methods for only one of our dependent measures (mean flow speed at user-chosen critical point locations). Additionally, we found that these two arrow-based methods exhibited similar patterns of performance across all of the task measures. Fig. 15 shows that these methods elicited below average (greater than 0.5 standard deviations above the mean) performance for all of the task measures, except advection response time for which all (but LIC) of the other methods performed similarly.

Fig. 15 shows that GSTR exhibited a consistent pattern of above average performance over all of the task measures. The good performance of GSTR on these tasks is interesting because it consists of integral curves seeded on a regular grid. While some sources suggest that the seeding will introduce biases into the visualization, those biases do not seem to have hindered performance in this study. Perhaps the fact that the streamlines are significantly longer than the grid spacing hides a bias that might otherwise be introduced.

Performance using OSTR was above average for the advection and critical point type identification tasks; the clear flow lines and directionality of the icons in OSTR probably contributed to users performing well in these tasks. However, the lack of icon density or speed information may have caused users to have difficulty using OSTR for the critical point locating task relative to performance on the other tasks (see Fig. 15).

On the other hand, the pattern of performance with LIC was almost the opposite of OSTR. Fig. 15 shows that user performance with LIC was well above average for the critical point locating task and below average for the advection and critical point identification tasks, with the exception of response time performance during the latter task. While LIC and OSTR share the visualization of clear streamlines in common, LIC does not display any flow direction information. This information is critical for determining particle advection and helps to disambiguate critical point types (i.e., users could quickly decide whether a critical point was a focus, saddle or node but were unable to accurately determine attraction versus repulsion). Conversely, the density of the streamlines in LIC provided clear points in the stimulus images where the critical points were located, facilitating performance for that task.

Fig. 15 shows that the LIT method elicited average user performance across most of the tasks relative to the other methods. Performance using LIT was above average for flow speed at user-chosen critical points and for response time for the critical point location task. These results seem to be consistent with the icons used in the LIT method showing flow speed (as the icons grow and shrink in the image) and critical point location (icon placement leaves distinct white areas at the critical point locations).

While the performance of specific methods is interesting, it is perhaps more valuable to look at common aspects of the methods that may explain good performance. Several factors seemed to be correlated with methods that performed well. First, methods that had a clear representation of the flow structure, as well as a good indication of the vector sign, supported consistently good performance across all task measures. This includes the OSTR, GSTR, and LIT visualization methods. Not surprising, methods that had a clear indication of critical point location performed well on the critical point location task. These included LIC, where critical points were indicated by the anomalous structure of the flow near the critical points, LIT, where the area of the wedge shapes shrinks to leave clear white regions around critical points, and GSTR, where the overlapping streamlines tend to cluster near the critical points.

Using information about which visualization features to include, it may be possible to modify or combine visualization methods to improve user performance. However, we caution against methods that directly support only the specific tasks described. These tasks are a part of the process of understanding a 2D vector field but there is also an overall understanding of the field that goes beyond them. This and other observations were taken from interactions with expert designers [26].

Performance of nonexperts and experts did not differ significantly. We saw some loose trends suggesting that experts might be slightly more accurate and slightly slower than nonexperts, but the differences were not statistically significant. We draw two inferences: first, the training at the beginning of the study was sufficient to acclimate the nonexpert users to the visualization methods and study tasks; and second, the more readily available nonexpert population may be effectively used for determining trends also seen in the expert population.

We carefully considered the methodology to use for our data analyses. We first decided to perform standard null hypothesis significance testing, calculating the omnibus

ANOVA for each task measure followed up by posthoc pairwise comparisons of the means. After creating our figures and weighing their utility for explaining the results, we decided to try a graphical analysis of the data using inferential confidence intervals [20]; these confidence intervals included a standard error estimator based on the normalized interaction variance for each condition [24]. These calculations allowed us to control for violations of the homogeneity of variance and covariance assumptions and also control the experimentwise Type I error at the nominal rate of 0.05. Using the inferential confidence intervals also allowed us to determine statistical differences between any condition means for each task measure by looking at the associated figure. We concluded that this method was optimal for conveying our results to the reader because it: 1) correctly controlled for statistical assumptions and experimentwise alpha level and 2) allowed the reader to quickly and easily determine the pattern and statistical significance of the results by observing the task measure figures.

Finally, we envision as a future project the extension of this work to 3D visualization techniques. It is not apparent that conclusions drawn within this work for 2D techniques extend naturally. For instance, in 3D, dense representations around critical regions my obfuscate other important features. Though somewhat frustrating, the move to 3D will probably require us to go back to the first principles of user studies and build a proper 3D study.

In summary, we have presented comparative performance results for three two-dimensional vector visualization tasks using six visualization methods. The choice of tasks clearly influences the results. The tasks seem representative of the types of tasks that fluids researchers want to perform from visualizations, although they could clearly be augmented. Our results show differences among the methods and suggest that the methods that attempt to visually represent integral curves, indicate critical points, and show flow directionality support better task performance.

ACKNOWLEDGMENTS

The authors would like to thank J.J. Braider and Morriah Horani for their helpful comments on the paper. They would also like to thank George Karniadakis and Peter Richardson for their expert input concerning fluid flow visualization and Jason Taylor for his help in verifying the calculations of the normalized per-condition mean-squared estimator. This work was partially supported by NSF (CCR-0086065). Opinions expressed in this paper are those of the authors and do not necessarily reflect the opinions of the US National Science Foundation.

REFERENCES

[1] E. Tufte, *The Visual Display of Quantitative Information.* Graphics Press, 1983.
[2] C. Ware, *Information Visualization: Perception for Design.* Morgan Kaufmann, 2000.
[3] W.S. Cleveland, *The Elements of Graphing Data.* Wadsworth, 1985.
[4] R. Kosara, C.G. Healey, V. Interrante, D.H. Laidlaw, and C. Ware, "Thoughts on User Studies: Why, How, and When," *Computer Graphics and Applications,* vol. 23, no. 4, pp. 20-25, July/Aug. 2003.
[5] V. Interrante, H. Fuchs, and S.M. Pizer, "Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture," *IEEE Trans. Visualization and Computer Graphics,* vol. 3, no. 2, Apr.-June 1997.
[6] R. Kosara, S. Miksch, H. Hauser, J. Schrammel, V. Giller, and M. Tscheligi, "Useful Properties of Semantic Depth of Field for Better f+c Visualization," *Proc. Joint Eurographics-IEEE TCVG Symp. Visualization 2002 (VisSym '02),* pp. 205-210, 2002.
[7] J.E. Swan II, J.L. Gabbard, D. Hix, R.S. Schulman, and K.P. Kim, "A Comparative Study of User Performance in a Map-Based Virtual Environment," *IEEE Virtual Reality 2003,* p. 259, Mar. 2003.
[8] D.H. Laidlaw, R.M. Kirby, T.S. Miller, M. da Silva, J.S. Davidson, W.H. Warren, and M. Tarr, "Evaluating 2D Vector Visualization Methods," *Proc. Visualization '01,* 2001.
[9] M.A.Z. Dippé and E.H. Wold, "Antialiasing through Stochastic Sampling," *Computer Graphics (SIGGRAPH '85 Proc.),* B.A. Barsky, ed., vol. 19, pp. 69-78, July 1985.
[10] R.M. Kirby, H. Marmanis, D.H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," *Proc. Visualization '99,* 1999.
[11] B. Cabral and L.C. Leedom, "Imaging Vector Fields Using Line Integral Convolution," *Computer Graphics (SIGGRAPH '93 Proc.),* J.T. Kajiya, ed., vol. 27, pp. 263-272, Aug. 1993.
[12] G. Turk and D. Banks, "Image-Guided Streamline Placement," *Proc. SIGGRAPH 96,* pp. 453-460, 1996.
[13] Mathworks, *Matlab.* Natick, Mass.: Mathworks, Inc., 1999.
[14] A. Globus, C. Levit, and T. Lasinski, "A Tool for Visualizing the Topology of Three-Dimensional Vector Fields," *Proc. Visualization '91,* pp. 33-40, 1991.
[15] S.E. Maxwell and H.D. Delaney, *Designing Experiments and Analyzing Data: A Model Comparison Perspective,* Belmont, Calif.: Wadsworth, 1990.
[16] D.H. Krantz, "The Null Hypothesis Testing Controversy in Psychology," *Am. Statistical Assoc.,* vol. 44, pp. 1372-1381, 1999.
[17] R.S. Nickerson, "Null Hypothesis Significance Testing: A Review of an Old and Continuing Controversy," *Psychological Methods,* vol. 5, pp. 241-301, 2000.
[18] R.P. Abelson, "On the Surprising Longevity of Flogged Horses: Why There Is a Case for the Significance Test," *Psychological Science,* vol. 8, pp. 12-15, 1997.
[19] R.L. Hagen, "In Praise of the Null Hypothesis Statistical Test," *Am. Psychologist,* vol. 52, pp. 15-24, 1997.
[20] W.W. Tryon, "Evaluating Statistical Difference, Equivalence, and Indeterminacy Using Inferential Confidence Intervals: An Integrated Alternative Method of Conducting Null Hypothesis Statistical Tests," *Psychological Methods,* vol. 6, no. 4, pp. 371-386, 2001.
[21] J.W. Tukey, *Exploratory Data Analysis.* Addison-Wesley, 1977.
[22] H. Wainer and D. Thissen, *A Handbook for Data Analysis in the Behavioral Sciences: Statistical Issues,* Graphical Data Analysis, pp. 391-458, 1993.
[23] M.E.J. Masson and G.R. Loftus, "Using Confidence Intervals for Graphically Based Data Interpretation," *Canadian Experimental Psychology,* vol. 57, no. 3, pp. 203-220, 2003.
[24] G.R. Loftus and M.E.J. Masson, "Using Confidence Intervals in Within-Subject Designs," *Psychonomic Bulletin and Rev.,* vol. 1, pp. 476-490, 1994.
[25] E. Batschelet, *Circular Statistics in Biology.* Academic Press, 1981.
[26] C. Jackson, D. Acevedo, D.H. Laidlaw, F. Drury, E. Vote, and D. Keefe, "Designer-Critiqued Comparison of 2D Vector Visualization Methods: A Pilot Study," *Proc. SIGGRAPH 2003 Sketches and Applications,* 2003.

David H. Laidlaw received the PhD degree in computer science from the California Institute of Technology, where he also did postdoctoral work in the Division of Biology. He is an associate professor in the Computer Science Department at Brown University. His research centers on applications of visualization, modeling, computer graphics, and computer science to other scientific disciplines.

Robert M. Kirby received ScM degrees in computer science and applied mathematics and the PhD degree in applied mathematics from Brown University. He is an assistant professor of computer science at the University of Utah's School of Computing and is a member of the Scientific Computing and Imaging Institute at Utah. His research interests lie in scientific computing and visualization.

Cullen D. Jackson received the PhD degree in experimental psychology from Brown University. He also received the BS degree in computer science from Trinity University. He is a postdoctoral research associate in the Computer Science Department at Brown University. His research interests include scientific visualization, human object and space perception, 3D user interactions, and intelligent interfaces.

J. Scott Davidson received the BA degree in computer science from Brown University in 2001. While at Brown he was involved in several research and education projects, including this vector field visualization user study. He is currently a software engineer at SensAble Technologies, Inc., in Woburn, Massachusetts.

Timothy S. Miller received the BS degree in cognitive science from Brown University. He is a research scientist in the Computer Science Department at Brown University. He is currently doing research in unobtrusive pen-based interfaces for note-taking with active assistance.

Marco da Silva received the BS degree in math and computer science at Brown University in 2001. He worked in Brown's Graphics Research Group as an undergraduate. He is currently working as a software engineer at Pixar Animation Studios.

William H. Warren is a professor and chair of the Department of Cognitive and Linguistic Sciences at Brown University. He studies human visual perception and visual control of action, including locomotion and navigation, using virtual reality techniques and dynamical systems modeling. He is the recipient of a Fulbright Research Fellowship, an NIH Research Career Development Award, and Brown's Teaching Award in the Life Sciences.

Michael J. Tarr is the Fox Professor of ophthalmology and visual sciences and a Professor of Cognitive and Linguistic Sciences at Brown University. He is interested in the human ability to visually recognize objects and has used psychophysical, computational, and neuroscientific tools to study this problem.