Evaluation of Multivariate Visualization on a Multivariate Task

Mark A. Livingston, Jonathan W. Decker, and Zhuming Ai

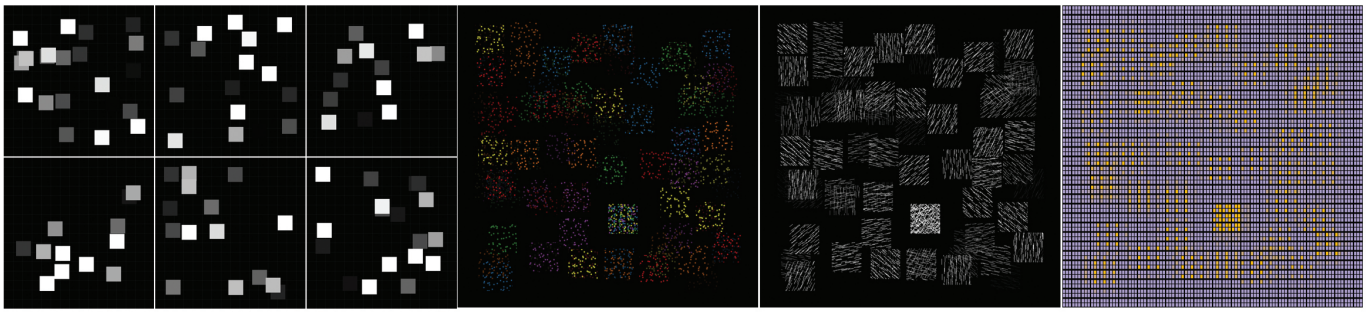

Fig. 1. Multivariate visualization methods evaluated on a task that was difficult to solve with the baseline visualization technique. From left: Juxtaposed Layers (the baseline case of spatially separate visualizations for each variable), Data-driven Spots, Oriented Slivers, and Attribute Blocks. The target patch is slightly below the center of the image.

Abstract—Multivariate visualization techniques have attracted great interest as the dimensionality of data sets grows. One premise of such techniques is that simultaneous visual representation of multiple variables will enable the data analyst to detect patterns amongst multiple variables. Such insights could lead to development of new techniques for rigorous (numerical) analysis of complex relationships hidden within the data. Two natural questions arise from this premise: Which multivariate visualization techniques are the most effective for high-dimensional data sets? How does the analysis task change this utility ranking? We present a user study with a new task to answer the first question. We provide some insights to the second question based on the results of our study and results available in the literature. Our task led to significant differences in error, response time, and subjective workload ratings amongst four visualization techniques. We implemented three integrated techniques (Data-driven Spots, Oriented Slivers, and Attribute Blocks), as well as a baseline case of separate grayscale images. The baseline case fared poorly on all three measures, whereas Data-driven Spots yielded the best accuracy and was among the best in response time. These results differ from comparisons of similar techniques with other tasks, and we review all the techniques, tasks, and results (from our work and previous work) to understand the reasons for this discrepancy.

Index Terms—Quantitative evaluation, multivariate visualization, visual task design, texture perception.

1 Introduction

One of the classic goals of scientific visualization is the presentation of large data sets. One early approach to solving the problem determined how to reduce the data sent to the rendering algorithm [4]; summary statistics indicated dominant patterns within the data. Such summaries may be presented through standard statistical graphics or via adaptations that maintain spatial layout [3]. Enabled by increasing graphics and display capabilities, other recent approaches have built on early work in glyph-based representations [1, 12, 13] to devise methods for presenting multiple variables simultaneously. Such techniques aspire to help the user to discern subtle patterns involving multiple variables, leading to analytical insights into the data.

One appealing aspect of multivariate visualization (MVV) techniques is that presentation of the detail within each of the data layers may lead the analyst(s) to unforeseen insights about the variables in each layer or their relationships between them. From such insights, new analytical processes may be derived and applied rigorously. On the other hand, integrated presentation of multiple values could lead to *information overload*, in which the analyst(s) cannot discern useful details, patterns, or insights because too much data is presented to permit reasoning about it. One long-term goal of our work is to determine the perceptual limits relevant to presenting multiple variables or data layers. The premise of this goal is that discovery of techniques that can lead to unforeseen insights will enable a wide range of analysis tasks to be performed without resorting to brute-force computation of all possible statistical evaluations.

Therefore, a number of MVV techniques have been developed; however, evaluation of the utility of these techniques remains underexplored in the literature. There are many good reasons for the lack of studies. User-based evaluations are not easy to design for comparisons of multiple, diverse visualization methods. Therefore, most techniques are evaluated with a single data set and task. While a focused evaluation may show the value of a technique, a comparison against other options would be valuable information for the visualization research community. Our fundamental goal in this work was to extend the set of evaluations that compare MVV techniques using the same task and data. Section 2 describes MVV techniques and past evaluations of them. We designed a new task and conducted our study (Section 3); we present the results and discuss the findings in light of previous work (Section 4).

2 Related Work

Research on MVVs benefits from a user-centered approach, encompassing both perceptual and cognitive studies of human capabilities. Guidelines for perceptual discernment of subtle differences could improve performance on a wide variety of data-intensive tasks using

- *Mark A. Livingston is with the Naval Research Laboratory, E-mail: mark.livingston@nrl.navy.mil.*
- *Jonathan W. Decker is with the Naval Research Laboratory, E-mail: jonathan.decker@nrl.navy.mil.*
- *Zhuming Ai is with the Naval Research Laboratory, E-mail: zhuming.ai@nrl.navy.mil.*

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

MVV techniques. We briefly review techniques and evaluations, with comments on our implementations.

2.1 Multivariate Visualization Techniques

Color Blending is perhaps the oldest and conceptually simplest MVV. Each variable is assigned a particular color; the value of each pixel is computed to be the weighted sum of the colors, with the weights derived from the data values. Thus the dominant hue of a pixel or region in visualization should indicate the greatest component value among the data values at that location. Each pixel is a visual sample, so the spatial resolution is equal to that of the display device, but expressing more than three independent variables through only three degrees of freedom requires a creative mapping. Even when displaying only three variables or with sufficient degrees of freedom in the display (printer or monitor, for example), perceptual limitations often interfere with the conveyed impression of data values.

Attribute Blocks build on early visualizations that use a cluster of shapes or a divided shape to represent multiple values at sample points [1, 12, 13]. Each attribute may be visualized with a continuous variable, such as color or intensity; variables are separated by their location within the cluster or shape [17]. Dynamically changing the array's configuration and the size and origin of the individual components enables synthesizing higher resolution than the initial sampling of multivariate data. Issues arise in determining how to sample the underlying data fields, since (unlike Color Blending), the multi-valued representation requires more than a single pixel to represent one sample. Thus rich features may be observed, but at a cost of the spatial resolution. If a data value is not constant over the cell assigned to that data layer, then an integration technique must be applied. We used an area-based technique; in retrospect, a nearest-neighbor or maximum-value strategy may have led to better results.

Oriented Slivers [22] encodes each data layer with short, grayscale lines on a randomly jittered grid. The orientation differentiates the data layers; the intensity encodes the data values. Sliver placement affects what frequency of the underlying data may be reliably understood. Further, using many slivers, wide slivers, or long slivers may prevent the user from distinguishing individual slivers. Still, the technique has the advantage of using few perceptually significant features, allowing the potential for many data layers to be visualized. We opted to restrict ourselves to the technique as defined [22] rather than invent extensions such as the use of color, although this is certainly an option.

Data-driven Spots [2] (DDS) is similar in spirit to pointillist art techniques, using the fact that the human visual system naturally fills space between samples. DDS encode each data layer with Gaussian kernels on a randomly jittered grid. The layers are differentiated by the size and hue, while intensity encodes the data value. Layers may also animate over the surface to further perceptual distance between them and to synthesize resolution beyond that created by the size and spacing of the spots, albeit perhaps by raising a conflict with the jitter pattern. As with Oriented Slivers, using many spots may affect the perceptible frequency of the underlying data. *Color weaving* [20] similarly works on the same concept of overlaying color on a high-frequency texture pattern. This technique does not rely on features such as Gaussian kernels, but on a color field at the display resolution. The closest analogy for DDS is non-overlapping, space-filling kernel sets, which is how we implement DDS. Color weaving has been shown to enable good performance in an evaluation, which is the topic of the next sub-section.

2.2 Evaluating Multivariate Visualizations

A few authors have conducted evaluations of MVV techniques with quantitative and qualitative studies and a variety of tasks, resulting in an assortment of observations. Height and density of vertical bars over a 2D domain were easily identified, but certain combinations with background elements (such as salience or regularity of samples in a dense field) made it hard to understand the data [7]. Brush Strokes (using color, texture, and feature hierarchies among luminance, hue, and texture) enabled verification [8] that perceptual guidelines for visualization [21] apply to non-photorealistic visualizations as well. Oriented Slivers [22] enabled users to perceptually separate layers within a data set. To get the best performance on identifying the presence of a constant rectangular target in a constant background field required a minimum separation of 15° between layers.

A key type of evaluation is testing task performance with MVV techniques. DDS enabled users to discern boundaries amongst as many as nine layers of data [2]. Other art-inspired techniques such as pointillism, speed lines, opacity, silhouettes, and boundary enhancement enabled users to track a feature over time more accurately and with a subjective preference [9]. Adding colors and altering texture properties such as line thickness or orientation in line-integral convolution created effective visualizations for multiple flow fields, as assessed by domain experts [20]. Ellipsoid glyphs were effective at showing tensor structure in diffusion tensor images, whereas layered Brush Strokes encoded field values and enabled users to understand relationships between layers, albeit with a potential for cluttered images [10]. This was not a serious problem in the task because the application displayed inter-dependent variables (data layers).

Other studies have compared multiple, diverse visualization techniques. The most relevant study for our work compared Color Weaving and Color Blending [5]. Users were able to read combinations of 2, 3, 4, and 6 data values with error rates between 7% (two values) and 17% (six values) with color weaving, whereas error rates were between 11% (two values) and 28% (six values) with color blending. Data values were encoded via single-hued color scales that varied jointly in saturation and luminance; users (sequentially) moved six sliders to indicate their responses. Another study [11] demonstrated that when a visualization explicitly represented a feature sought – e.g. showed the sign of vectors, integral curves, and critical point locations – users localized these features more accurately. Experts and non-experts did not show significant differences. Multi-layer texture synthesis enabled users to perform with no significant difference from Brush Strokes for weather data visualization [19].

In a previous study, we found [16] that the parameterized patterns of DDS and Oriented Slivers helped users perform critical point (maximum) detection more accurately and faster than glyph representations of Brush Strokes and Stick Figures and more accurately than Color Blending. We also found some techniques were sensitive to monitor settings (brightness and contrast) and room lighting conditions. On a trend detection task [14], DDS and a baseline case of separate grayscale visualizations outperformed Brush Strokes, Dimensional Stacking, Oriented Slivers, and Color Blending with respect to accuracy, but not with respect to response time. A follow-up study [15] found that the technique of Attribute Blocks improved greatly over Dimensional Stacking, but that adjustments to the DDS technique expected to improve performance (via improved contrast) worsened user performance. Previous exposure to techniques lowered response time and subjective workload, but not error on the trend localization task.

3 Study Design

We describe our experimental task, then discuss the independent variables we studied and the dependent variables through which we explored these factors. Our hypotheses were informed by our previous studies. We present statistical results in this section, with discussion to follow in the next section.

3.1 Experimental Task

We found in a trend detection task that the baseline technique – presenting variables in spatially separated grayscale visualizations – performed as well or better than the integrated MVV techniques [14, 15]. For detection of the maximum in a single variable [16], it seems obvious to think that detection of the maximum would best be accomplished when looking only at that variable (though we did not test the condition). So one goal in designing the task for this study was for the task to show the value of MVV techniques over the baseline condition.

We built on the task used to study DDS in the original exposition of the technique [2]. The original study asked users to estimate the percentage of a shape (binary data) presented in one target layer overlapped by the shape in a second target layer. It then asked users to

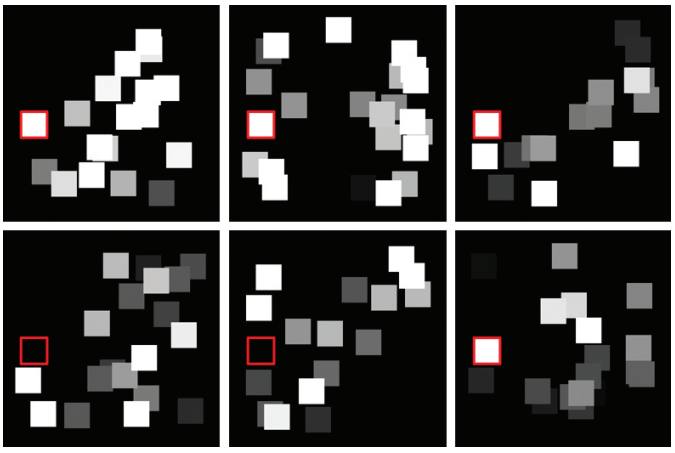

Fig. 2. This training image was shown to the user to explain their task. Users were shown that the target (outlined here in red as it was for the training) did not overlap any distractors in any variable and was at the maximum value (white for the baseline visualization) in the layers that were part of the target and at the minimum value (black for all techniques) in other layers.

sketch the region of overlap between these two shapes. Between zero and seven other layers presented a shape that was meant to distract the user from the two target layers. DDS enabled better performance (on both tasks) than side-by-side presentation of the targets (which had no distractors); this was true for any number of distractors present in the DDS visualization.

We conceived of the task as having a user determine the region with the greatest number of variables that were overlapping at their maximum value. With the MVV representations we use, the perceptual response comes from finding the area of maximum texture density. We chose to present six variables (data layers) in every stimulus (Figure 2 shows the training image for the baseline visualization), with the target consisting of four, five, or six variables with three pre-defined sizes. For each combination of number of target layers and target size, we generated a target location within the domain[1]. We then assigned the target to the appropriate number of layers (setting those layers to the maximum value and other layers to zero). Next, we added distractors to the domain. These were essentially targets involving fewer layers, but with two important differences. First, the target never overlapped with any distractor, but distractors could overlap each other. Second, the distractors had at least one more layer completely empty than the target had (e.g. if the target had five variables, no more than four layers were used for a distractor), and if only one more layer was empty in a distractor, then at least one other layer was limited to half the maximum value. This constraint is perhaps easiest to conceive as being placed on the sum of the variables in the target and distractors. If each variable had a range of $[0..1]$, then the target had a sum s of four, five, or six, whereas the distractors had a sum of $[0..(s-1.5)]$.

3.2 Independent Variables

The above discussion mentioned some independent variables in passing. We used four MVV techniques: DDS, Oriented Slivers, Attribute Blocks, and a baseline condition we call Juxtaposed Layers (a grid of separate images for each variable, as in Figure 2). Figure 3 shows examples of what data look like with the integrated techniques. Figure 4 shows the legends for the integrated techniques; these legends were present during the trials. This was the independent variable of primary interest in our study. Since one of our long-term goals is to determine how many variables may be comprehended, the secondary independent variable was the number of variables (data layers) included in the target: four, five, or six. To help understand the difficulty of the task, we used three target sizes: 31, 61, and 91 pixels; the target was

[1]See the supplemental material for the stimuli data files.

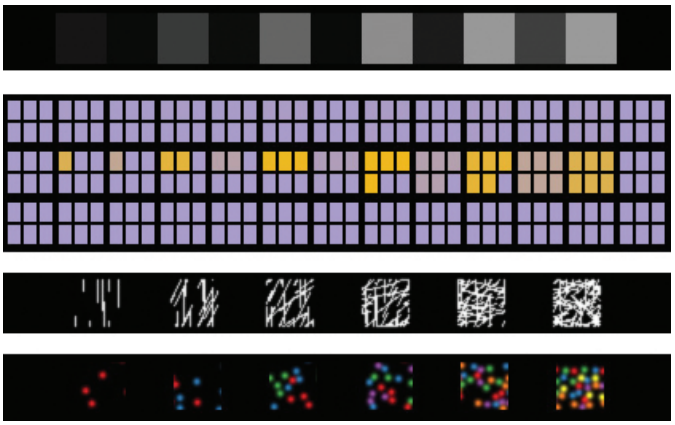

Fig. 3. Example data as represented by the integrated MVV techniques. The top shows grayscale squares indicating field values (sums); the second block shows these values with Attribute Blocks, the third block shows Oriented Slivers, and the last block shows DDS.

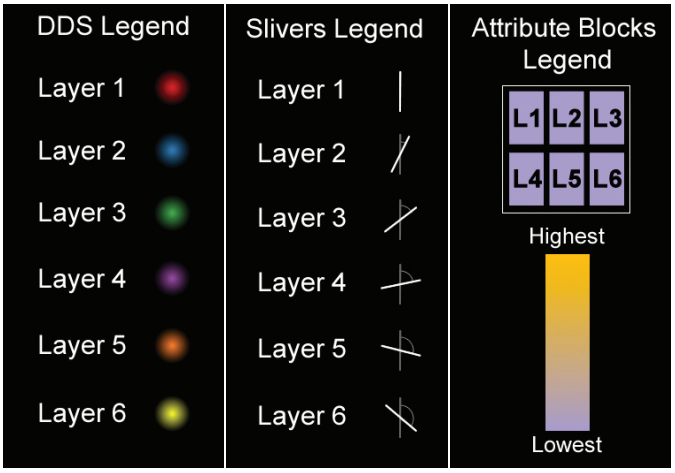

Fig. 4. The technique legends helped the users understand how the six variables were presented in the three integrated MVV techniques. From left to right: Data-driven Spots, Oriented Slivers, and Attribute Blocks.

centered on a pixel, and the odd number made the target generation easier. These sizes represented 3%, 6%, and 9% of the 1024-pixel domain square, respectively. Because of the space requirements, the juxtaposed layers were shown in 512×512 images, separated by a small white space. In analyzing the results in this paper, we also consider whether user experience and repetition were contributing factors to performance as independent variables.

3.3 Dependent Variables and Hypotheses

We measured the error with respect to target value. Specifically, the error was the value (number of overlapped layers) at the target minus the number of layers at the selected location. Since the maximum target value was six, this measure has (in theory) a range of [0,6]. We also measured response time and the number of times a user changed an answer; the users were informed that they could change their answer as many times as they wished. Response time was measured from the onset of the stimulus until the time of the selection of the final answer. Finally, we measured the subjective workload associated with each technique through the NASA Task-load Index [6]. We formulated the following hypotheses (using $\alpha = 0.05$) based on previous results from our own work as well as the literature.

1. We expected all the integrated MVV techniques to outperform the baseline case of Juxtaposed Layers, demonstrated by lower

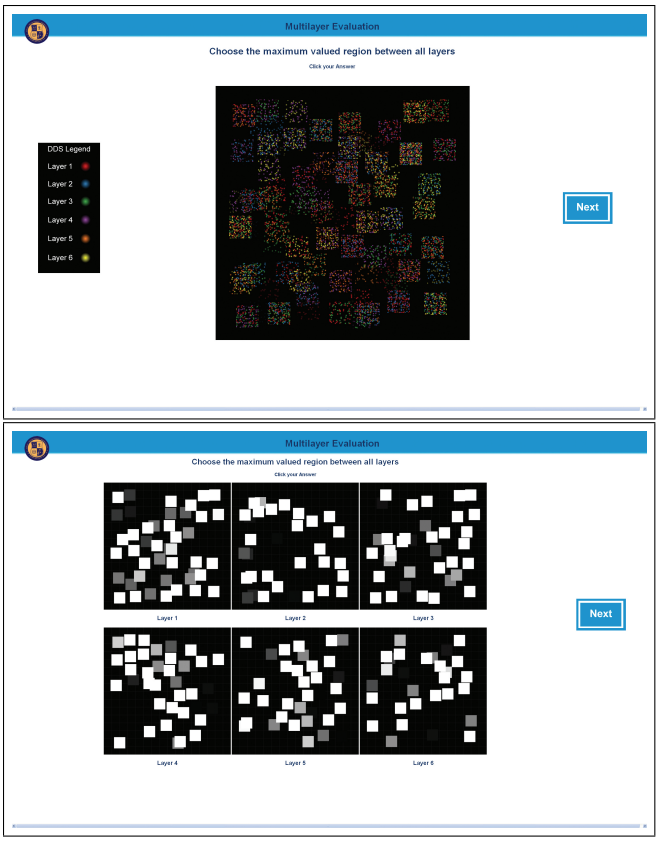

Fig. 5. Screen-captured images for the (*top*) Data-driven Spots and (*bottom*) Juxtaposed Layers techniques.

error, faster response time, and selection of fewer answers within each trial.

2. We further expected DDS to outperform Oriented Slivers and Attribute Blocks for error.

3. We expected the error to increase with increasing number of target layers.

4. We expected the error to increase with decreasing target size.

5. We expected (based on our previous study) that user experience with the techniques would lead to faster response time.

3.4 Subjects and Procedures

The control software was implemented as a set of web pages viewed with the Google Chrome browser (version 17.0.963.83m) under WindowsXP (Service Pack 3). The user sat at a standard desktop workstation and viewed the stimuli on a 30-inch Dell WFP3008 monitor running at 2560×1600 resolution. Factory default settings were maintained for brightness (75), contrast (50), sharpness (50), gamma ("PC"), color settings mode ("Graphics"), and Preset mode ("Desktop"). The room had standard fluorescent lights. We did not enforce a precise viewing distance; the desktop yielded a viewing distance of 67cm for a typical seated position (giving pixel pitch of 0.25mm). Figure 5 shows images of the entire data trial screen. This configuration is identical to the configuration in our previous studies [16, 14, 15].

Twelve subjects (8 male, 4 female) participated in the study; they averaged 38 years of age. All self-reported having normal or corrected-to-normal visual acuity and normal color vision. All reported being heavy computer users; five had not seen the integrated visualizations techniques previously; the remaining subjects had participated in one (four users), two (one users), or three (two users) of our previous studies. The subject first read a set of instructions about the task, and was then given the hints about the target (no overlap with distractors, binary values for the overlapping layers). The subject then proceeded through each technique. The order of visualization techniques was determined by a Latin square balanced for first-order residual effects [23]. Each technique began with instructions specific to the technique. The subject then completed two practice questions, in which only one answer could be selected, but the correct answer was shown. This was followed by the data trials; the order of trials within each technique were determined by random permutation. At the end of each technique, the user completed the NASA TLX. Each subject completed four repetitions of the combination of target size and number of target layers for each of the four visualization methods, for a total of $4 \times 3 \times 3 \times 4 = 144$ data points per subject (1728 total).

3.5 Study Results

We ran a series of repeated measures ANOVA calculations to determine statistically significant effects[2]. Our first goal was to verify Hypothesis 1, that the integrated visualization techniques would outperform the baseline visualization of separate juxtaposed grayscale visualizations of the layers.

3.5.1 Outperforming the Baseline

Using a 4 (MVV Technique) × 3 (Target Layers) × 3 (Target Size) repeated-measures ANOVA, we found main effects of MVV Technique on error, response time, and number of answers selected (Table 1). Juxtaposed Layers led users to significantly greater error, slower response times, and more answers selected than the other techniques. See Table 2 for the means and standard deviations in each of the dependent variables. These results clearly support our first hypothesis, and they are consistent with the result for DDS on the two-layer overlap problem. Juxtaposed Layers yielded almost three times the error of the second-lowest performing technique (Attribute Blocks), required approximately 4.5 times the response time, and was the only technique where users were basically forced to explore multiple answers to arrive at their final choice. These differences also appear to be reflected in the subjective workload ratings; the baseline technique was rated as having a significantly higher workload.

3.5.2 Relative Performance of Integrated Techniques

Our second hypothesis was that DDS would exceed the performance of Oriented Slivers and Attribute Blocks. We conducted post-hoc Welch's t-tests to look for significant differences between the three integrated MVV techniques. We found that users were far more accurate with DDS than the other two integrated techniques. This result supports Hypothesis 2, that DDS would perform best among the integrated MVV techniques with respect to error. We did find some significant differences between the three integrated MVV techniques on response time and on number of answers selected (Table 1), but these are small and not meaningful for us at this time. We did find a curious result of a main effect of MVV technique on subjective workload: users rated DDS as having *higher* workload than the other two integrated techniques. Figure 6 shows the results for error, response time, number of answers, and subjective workload.

3.5.3 Effect of Number of Target Layers

We turn now to the other independent variables; however, we restrict our analysis to the integrated techniques only. Given the results stated above for the baseline technique, we are not interested in its effect on the other independent variables. The number of data layers in the target had a main effect on error and response time, but not on the number of answers selected (Table 1). However, the results do not quite support Hypothesis 3. Having six layers overlap for the target caused the greatest error, but having four layers was the next most difficult case, not five. Thus we could have supported a hypothesis

[2]Results from all ANOVAs conducted appear in the supplemental material. We used Pipe-Stat [18] to conduct ANOVAs. Post-hoc t-tests and correlations were conducted with online calculators at www.danielsoper.com and www.wessa.net.

Table 1. Hypothesis tests used to reach conclusions for the dependent measures. Sharp readers will notice that one subject's TLX ratings were discarded due to not assigning any work to any of the factors, giving all techniques a workload rating of zero. Degrees of freedom for the t-tests are clamped at two less than the number of cases yielded by the study design, in which all factors were crossed for error, response time, and answers.

Test and Factor(s)	Error	Response Time	Number of Answers	Workload
ANOVA: MVV Technique	$F(3,33) = 32.65, p = 0.00$	$F(3,33) = 35.48, p = 0.00$	$F(3,33) = 45.57, p = 0.00$	$F(3,30) = 19.20, p = 0.00$
t-test: DDS vs. Slivers	$t(106) = 6.59, p = 0.00$	$t(106) = 3.14, p = 0.002$	$t(106) = 0.65, p = 0.52$	$t(9) = 1.33, p = 0.22$
t-test: DDS vs. Att. Blocks	$t(106) = 5.20, p = 0.00$	$t(106) = 3.23, p = 0.002$	$t(106) = 2.54, p = 0.01$	$t(9) = 1.05, p = 0.32$
ANOVA: Num. of Layers	$F(2,22) = 7.45, p = 0.003$	$F(2,22) = 5.37, p = 0.01$	$F(2,22) = 1.80, p = 0.19$	
ANOVA: Target Size	$F(2,22) = 89.92, p = 0.00$	$F(2,22) = 8.98, p = 0.001$	$F(2,22) = 4.30, p = 0.03$	
ANOVA: Experience	$F(1,10) = 1.83, p = 0.21$	$F(1,10) = 6.17, p = 0.03$	$F(1,10) = 0.06, p = 0.82$	
ANOVA: MVV-by-Num. Layers	$F(4,44) = 8.79, p = 0.00$	$F(4,44) = 9.06, p = 0.00$		
ANOVA: MVV-by-Target Size	$F(4,44) = 35.64, p = 0.00$	$F(4,44) = 3.15, p = 0.02$		

Table 2. The mean and standard deviation for each MVV technique for each of the three objective dependent measures and the subjective workload rating shows the difficulty users had in attempting to complete the task with the baseline technique. Error is expressed in units of layers (range: 0-6), time in seconds, answers in a count, and workload through NASA TLX.

Name	Error (layers)	Error Std. Dev.	Time (sec)	Time Std. Dev.	Answers (count)	Answers Std. Dev.	Workload (TLX)	Workload Std. Dev.
JuxLayers	1.54	1.03	46.72	36.44	8.87	8.09	65.53	15.40
DDSpots	0.09	0.21	7.33	6.01	1.09	0.28	40.74	23.17
Slivers	0.42	0.48	9.62	4.25	1.11	0.24	27.86	22.38
Attrib	0.47	0.73	9.74	5.27	1.20	0.37	31.79	16.24

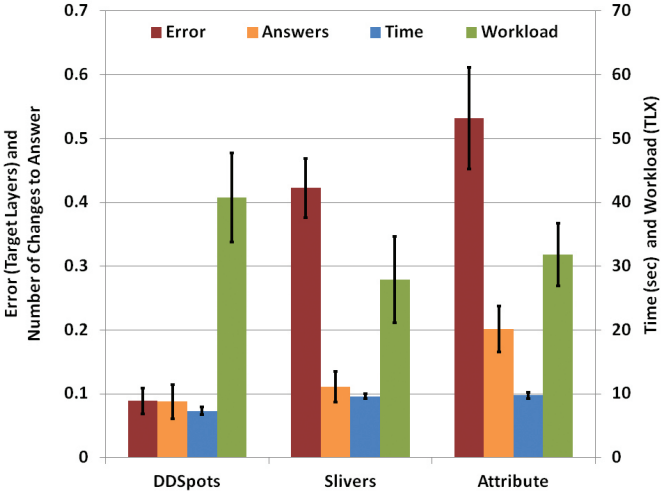

Fig. 6. Graph of dependent measures for the integrated MVV techniques. There was a main effect on error (red) and workload (green), but not on time (blue) or answers selected (orange). Error and number of changes to answers (i.e. one less than the number of answers selected, to align the graphs better) are on the primary axis on the left. Response time and workload are on the secondary axis on the right.

that stated only that six layers would be the most difficult and not predicted a complete ordering with respect to increasing number of layers. Further, we found that users were 13% *faster* with six layers in the target than with five layers, which is a bit counter-intuitive and a result that we shall discuss in Section 4.

3.5.4 Effect of Size of Target

The size of the target had a main effect on error, response time, and the number of answers selected. However, the results again do not support the complete ordering predicted in Hypothesis 4. The smallest target size was clearly more difficult, but there was no significant difference between the two larger sizes. Users were fastest with the largest target size, with a small but not significant difference between the smallest and middle sizes. Users changed their answers at a slightly increasing rate with decreasing target size.

3.5.5 Effect of User Experience

We expected (based on our past work) to see users who had participated in previous studies perform faster. We found a main effect of (binary) user experience on response time (Table 1). Returning users were on average 30.7% faster than subjects who were participating in our sequence of studies for the first time. This confirms Hypothesis 5.

3.5.6 Other Findings

We found significant interactions between MVV Technique and the number of target layers for error and for response time. We found significant interactions between MVV Technique and the target size for error and for response time. Since these results (Table 1) give us insight into the usability of the techniques and also implicitly show the main effects of the number of target layers and of the target size, we graph these results in Figure 7.

There was a significant interaction between the number of target layers and the target size for error – $F(4,44) = 5.128, p = 0.002$. For all number of target layers, the smallest targets were most difficult, but the magnitude of the increase in difficulty from the middle size down to the smallest size was quite a bit lower for five layers than would be expected looking at the jumps for four and six layers.

We checked whether fatigue had an effect on error by running a 3 (MVV Technique) $\times$ 36 (Count) ANOVA with the MVV technique and the count of questions as factors; we found no significant effect of the count of questions completed – $F(35,385) = 0.798, p = 0.789$. Similarly, we conducted a 3 (MVV Technique) $\times$ 3 (Target Layers) $\times$ 3 (Target Size) $\times$ 4 (Repetition) ANOVA to see if repetition of the combination of target size and number of target layers had a main effect; we found no significant effect – $F(3,33) = 0.860, p = 0.472$. Analogous ANOVA calculations revealed that there was no significant effect of trial count or repetition on the number of answers selected.

We ran a filter on the error to find trials where the response was judged to be incorrect, but the error in pixels from the correct answer was smaller than the size of the target. There were only nine such errors in 494 trials that saw errors (out of 1728 total trials), so we cannot attach statistical significance to the occurrence of such an event. But we do find it curious to note that of the nine such errors, seven saw selections that were no more than seven pixels away from the target patch – and all of these were trials with Attribute Blocks and the smallest target size (31 pixels). (Two trials saw selections that were almost the size of the target patch – 61 or 91 pixels, respectively – with Oriented Slivers.)

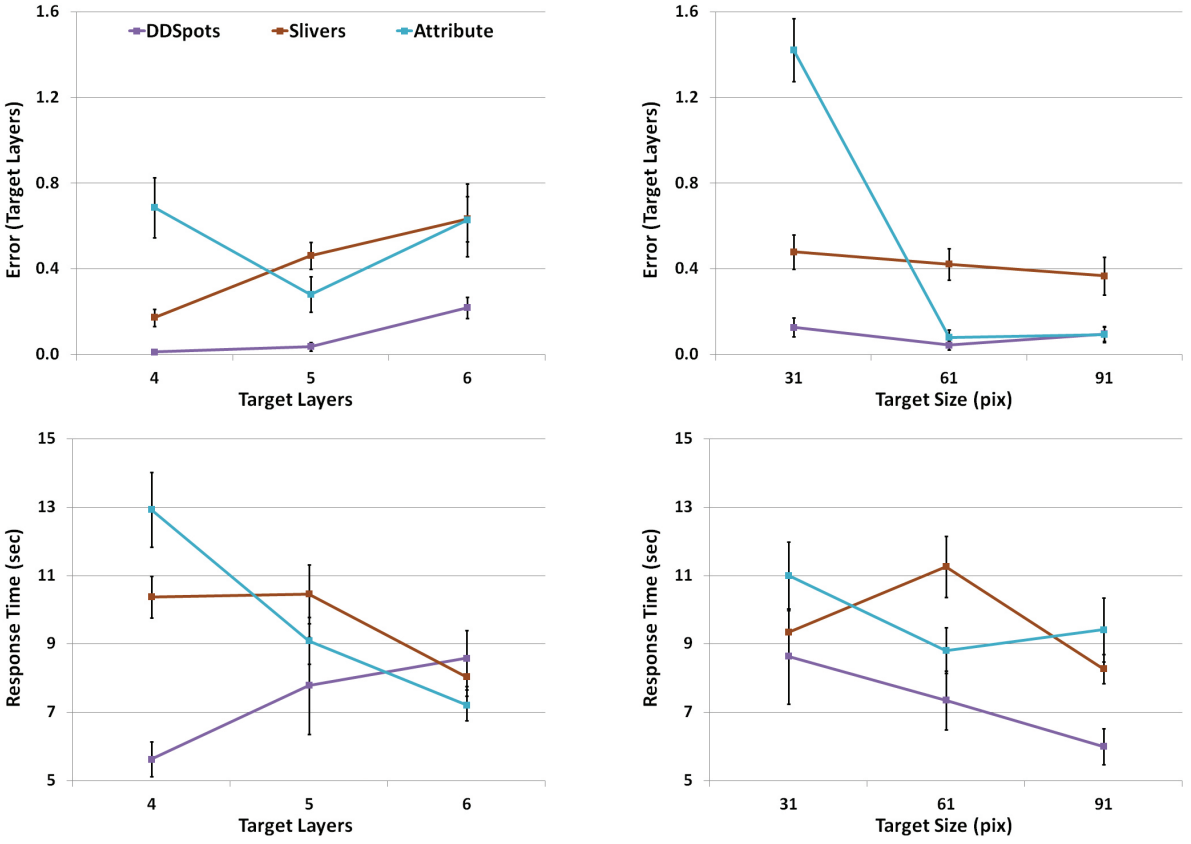

Fig. 7. Significant interactions were found between MVV technique and number of target layers for both error (top left) and response time (bottom left). Significant interactions were also found between MVV technique and target size for both error (top right) and response time (bottom right). Error bars in all graphs indicate one standard error unit.

We found a significant three-way interaction between MVV technique, number of target layers, and target size. While complex, it shows that for Attribute Blocks, the target size was the critical element – the small one was really hard (no matter how many layers), but for Oriented Slivers, the number of layers was what mattered (five and six were hard), and for DDS, it was more about number of layers, but it took six layers to make it hard.

4 Discussion

We find several of our results to be interesting in light of our own previous work and work reported elsewhere in the literature. The overlap task was inspired by chemical assays, in which (simultaneous) presence of elements is of interest. This task could be performed well only with the integrated MVV techniques, albeit with unequal success across the sub-cases we presented to users. Fundamentally, it appears that the density of symbols was the key perceptual tool on which users relied in order to solve the task[3]; texture density has been shown to be useful in previous studies. Healey and Enns [7] reported that shorter, denser, and sparser targets were harder to identify than taller targets in 3D perspective texture fields; background density interfered with searching for those three target types, and height interfered with searching for denser targets. Further, color interfered with finding denser targets. The applicability of these results to our 2D visual representations is open to question. With the independent variable of number of target layers, we tested far more values of density. Ware [21] notes that resolvable size (inverse of frequency) difference for Gabor texture patterns is about 9% change.

Looking at the stimuli for the integrated techniques, we can measure the density of the target to support the assertion that density was the cue our users employed. For each integrated MVV technique, we counted the number of pixels that were "on" within the target region. We defined "on" for DDS and Oriented Slivers (Figure 9) as any pixel that was of intensity at least 0.25 (conceiving range as [0..1]), whereas for Attribute Blocks, we defined "on" as a heat map value at least one quarter of the distance from the minimum value to the maximum value. (More precise definitions may assist with comparisons across techniques; for now, we seek only to understand each technique separately.) We found modest *positive* correlations (Figure 8) between the density and the error rate for DDS (0.37) and Oriented Slivers (0.61), but a modest negative correlation for Attribute Blocks (-0.17). This may seem counter-intuitive: the performance would be expected to improve (i.e. error decrease) with increasing density of the target. However, if we measure the ratio of density of the target to the most-dense distractor, we see the trend we expect. *Contrast in density* correlated negatively with user error for DDS ($R = -0.28, t(34) = 1.726, p = 0.09$) and Oriented Slivers ($R = -0.51, t(34) = 3.450, p = 0.002$), but positively for Attribute Blocks ($R = 0.57, t(34) = 4.097, p = 0.000$). While there were exceptions, which may be due to proximity of the distractions to the target and the sampling issue for Attribute Blocks discussed below, this does appear to offer an explanation for when and how well these techniques assisted the users.

We are pleased to note that the baseline case of presenting separate grayscale visualizations proved to be a poor interface for solving this task. The baseline case had fared so well in previous tests that one could reasonably question the utility of integrated visualizations. We assisted the comparison of locations in the Juxtaposed Layers by drawing a background grid in each layer's image, so that the task of determining whether a square was in the same location in four or more layers was easier. Still, we have a conclusive demonstration that the baseline case will not support this task. At the same time, we ac-

[3] We credit the reviewers for this insight.

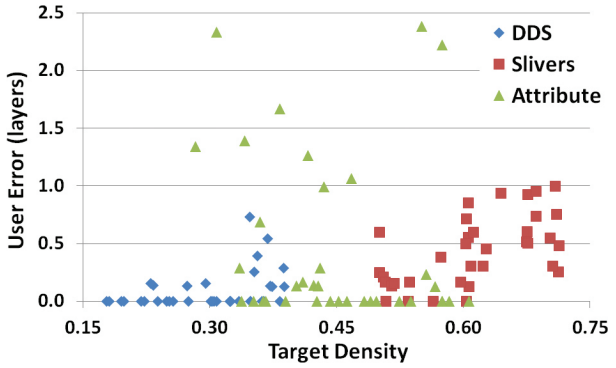

Fig. 8. A scatterplot of user error versus target density for each trial question (averaged over all users) shows that merely increasing density did not improve performance. Not shown is the ratio of density between target and distraction. The ratio appears to offer a better explanation for improved performance.

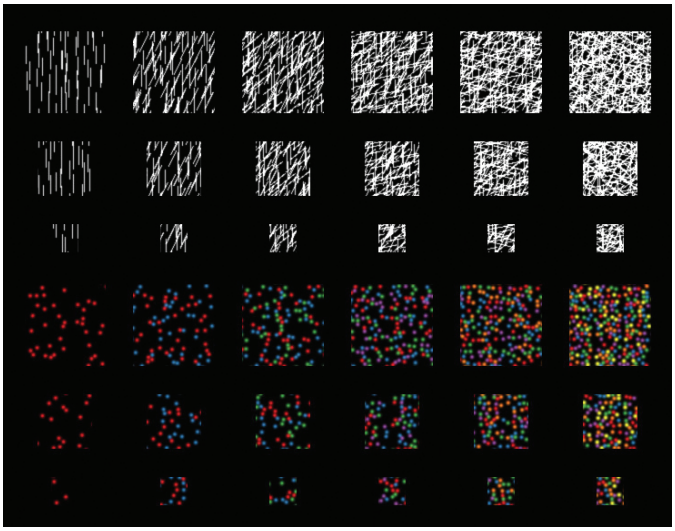

Fig. 9. Sample patches with (from left) one to six layers for Oriented Slivers (top three rows) and DDS (bottom three rows). For each technique, the three target sizes are shown.

knowledge that there is a design space in the task itself that we have not explored that could affect the difficulty. First and foremost is the separation (in number of layers present) of targets and distractors (we used 1.5 "max-value" layers of separation). The number of distractors clearly will have an effect (we generated 80), and the degree of overlap between the target and distractors (which we prevented) and distractors and other distractors (which we permitted entirely) will also. We explored the number of target layers and the target size to the small extent possible in a short user session (and these results are discussed above from a statistical standpoint and below for meaning). Thus we can see many opportunities for follow-up studies in these directions.

We started this line of research with a curiosity about the number of variables that could be comprehended with various MVV techniques. DDS continues to perform extremely well, and generally better than other MVV techniques. Based on the evidence of target density, one could hypothesize that because our implementation of DDS used a space-filling algorithm for the layout of the kernels, DDS had an advantage in this task. The Oriented Slivers would not appear to lend itself to such an implementation, and in fact, our current implementation tends to create "star-burst" patterns in which clusters of slivers intersect at a central point. Perhaps an artful design could be devised as a space-filling map of slivers for a small number of data layers. This could then enable a better test of this hypothesis than we can with our current implementation of Oriented Slivers. DDS and Attribute Blocks both lend themselves to space-filling implementations, as does Color Weaving. In fact, these three techniques can be implemented in a way that creates similar visualizations. The big difference between our implementation of DDS and Attribute Blocks is the sampling pattern.

We believe sampling explains the poor performance of Attribute Blocks. To illustrate this, we present the cases in which users missed the target by one to seven pixels (Figure 10). The five trials involved (two cases at the upper left – with nearly co-incident user selections, two at the upper right, and one in the other trials) all have targets which span multiple sampling cells used by Attribute Blocks to compute value for the block. This led to the target often having the appearance of two nearby cells having modest values, especially at the smallest target size, which was smaller than the Attribute Block size. While we could fix this for this case by making the Attribute Blocks (and thus the sampling cells) smaller, this is not a general solution; there will always be a smaller target than we can display. Varying this sampling pattern may improve the performance of Attribute Blocks. In addition, the heat map used in our implementation of Attribute Blocks has lower resolution than the black-to-saturated colors in DDS (and than Color Weaving would use); this color resolution appears to conspire with the sampling pattern to challenge our users.

One may posit that Juxtaposed Layers performed poorly because each variable was presented with half the spatial resolution of the integrated techniques. We provide evidence to dispel this notion by looking at the error for Juxtaposed Layers with only the largest target size versus the smallest target with integrated techniques. The largest target in Juxtaposed Layers is approximately 50% bigger than the smallest targets for the integrated techniques, so this should remove size from consideration as a reason. Table 3 shows that even with the benefit of *larger* size, Juxtaposed Layers was still significantly worse (using Welch's t-test) than DDS and Oriented Slivers. Having discussed the particular challenge for Attribute Blocks at the smallest size above, we find this evidence convincing that Juxtaposed Layers was not unfairly compared due to the spatial resolution for each individual variable.

Table 3. Statistical analysis of Juxtaposed Layers at the largest size target, which was scaled by one-half in the presentation of each layer (so 45 pixels) versus the integrated MVV techniques at the smallest target size. For each integrated technique, Welch's t-test is conducted. DDS and Oriented Slivers were significantly better at these target sizes. Each row has N=36 cases; size is in pixels; StdDev = standard deviation.

Technique	Size	Mean	StdDev	t-test vs JuxLayers
JuxLayers	91/2	1.40	1.07	
DDSpots	31	0.13	0.26	t(34)=6.91, p=0.00
Slivers	31	0.48	0.48	t(34)=4.69, p=0.00
Attribute	31	1.24	0.80	t(34)=0.72, p=0.47

Regarding user experience, we believe users who have experience with the techniques believe they have a strategy that works. Since there is no feedback, there can be no correction to any errors in this strategy. This would explain why they get faster, but not more accurate, than first-time users. We have timing data for how long people studied the instructions and the tutorial questions, but with twelve users, it is quite sparse for drawing inferences. Perhaps by combining it with previous studies we can draw some insights. It may be interesting to give subjects feedback during data trials to see what changes occur in the results for error and response time as a function of user experience.

Understanding the interaction between the number of target layers and the target size required some digging into the data. With our previous task, we created a balanced factor in the follow-up study to examine how the distance between the target and the nearest distractor affected the performance. It appears this may be wise with our new task as well. Consider the images for each technique that appear in the teaser image; the target (just below the image center) is somewhat isolated from the distractors. It turns out that for each combination of number of target layers and target size, this distance was relatively

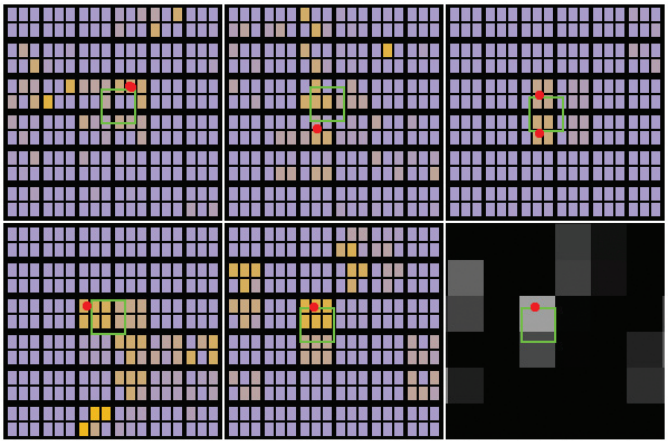

Fig. 10. Trials of Attribute Blocks where the user's response (red dot) was outside the target (green square) by one to seven pixels. The grayscale version of the bottom center trial appears at the lower right; one can see the sampling issue, where the target spans two of the large sampling blocks for Attribute Blocks, lowering the value by the averaging process and leading the user to interpret the more golden sampling block as the target location.

evenly split between near, medium, and far distance rankings (by our own subjective judgment). The only case for which this was not true was the case of five-layer targets at the smallest size; they were always somewhat isolated from the distractors (and never "near" distractors). This appears to have made the task easier and caused the interaction. With this knowledge in mind, we could create a formal metric for the distance and run a regression to see if it seems to explain the performance. In a follow-up study, we would plan to generate stimuli that vary this metric in a controlled fashion as an independent variable.

5 Conclusions and Future Work

Our results lead us to conclude that our implementation of DDS had the most advantageous design, (nearly) filling the visualization in the case of all variables at their maximum value. We had hoped that users would be forced to attend to all variables, which we believe is the most interesting test for MVV techniques. We will have to consider alternate tasks (which is always useful). Our goal is not merely to determine the best visual representation to accomplish a particular task; we again caution that if a task were known to be the proper analysis task, we would argue for directly computing the result and displaying that single image to the user. The argument for these types of evaluations is that the proper analysis technique is not always known ahead of time. Thus an understanding of which techniques will lend themselves to perceiving which types of data patterns is important knowledge.

As with any user study, there are any number of parameters in the study design we could explore further. Certainly, we could include more MVV techniques, increase the number of variables or expand the range of target sizes. Each of the techniques has a design space that we could explore to refine the individual techniques. We used parameters that we believed would most benefit each technique, based on our past studies; however, we do not claim that any of the MVV techniques is optimal for this task with regard to its own design space. However, faced with a choice to refine each technique separately with no insight into how it would compare against other techniques, we would likely waste effort on techniques that were limited. Thus we opted for a broad-based study. With the knowledge learned here, a refinement study for each of the various techniques could have merit.

We are curious about the evidence for increased speed without increased accuracy with the techniques. This is an area that deserves further exploration. The ultimate goal of these techniques is to make exploration of the data more efficient *and* more effective. While either alone is appreciated, we wonder what training might make these techniques truly effective and efficient tools for data exploration.

Acknowledgments

The authors wish to thank the anonymous subjects and the anonymous reviewers for their time and helpful suggestions. This work was supported by the NRL Base Program.

References

[1] J. Beddow. Shape coding of multidimensional data on a microcomputer display. In *Proceedings of IEEE Visualization*, pages 238–246, Oct. 1990.
[2] A. A. Bokinsky. *Multivariate Data Visualization with Data-driven Spots*. PhD thesis, The University of North Carolina at Chapel Hill, 2003.
[3] D. B. Carr and L. W. Pickle. *Visualizing Data Patterns with Micromaps*. CRC Press, 2010.
[4] H. Hagen. *Visualization of Large Data Sets*, chapter 12, pages 187–198. Academic Press, 1994.
[5] H. Hagh-Shenas, V. Interrante, C. Healey, and S. Kim. Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. In *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*, page 164, 2006.
[6] S. G. Hart and L. E. Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In P. A. Hancock and N. Meshkati, editors, *Human Mental Workload*, pages 239–250. Elsevier Science Publishers, 1988.
[7] C. G. Healey and J. T. Enns. Building perceptual textures to visualize multidimensional datasets. In *IEEE Visualization*, pages 111–118, 1998.
[8] C. G. Healey, L. Tateosian, J. T. Enns, and M. Remple. Perceptually based brush strokes for nonphotorealistic visualization. *ACM Transactions on Graphics*, 23(1):64–96, 2004.
[9] A. Joshi. *Art-inspired techniques for visualizing time-varying data*. PhD thesis, The University of Maryland, Baltimore County, 2007.
[10] D. H. Laidlaw, E. T. Ahrens, D. Kremers, M. J. Avalos, R. E. Jacobs, and C. Readhead. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings of IEEE Visualization '98*, pages 127–134, 1998.
[11] D. H. Laidlaw, R. M. Kirby, C. D. Jackson, J. S. Davidson, T. S. Miller, M. da Silva, W. H. Warren, and M. J. Tarr. Comparing 2D vector field visualization methods: A user study. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):59–70, January/February 2005.
[12] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring N-dimensional databases. In *Proc. of IEEE Visualization*, pages 230–237, Oct. 1990.
[13] H. Levkowitz. Color icons: Merging color and texture perception for integrated visualization of multiple parameters. In *Proceedings of IEEE Visualization*, pages 164–170, 420, Oct. 1991.
[14] M. A. Livingston and J. W. Decker. Evaluation of trend localization with multi-variate visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2053–2062, Dec. 2011.
[15] M. A. Livingston and J. W. Decker. Evaluation of multi-variate visualizations: A case study of refinements and user experience. In *SPIE Visualization and Data Analysis*, Jan. 2012.
[16] M. A. Livingston, J. W. Decker, and Z. Ai. An evaluation of methods for encoding multiple, 2D spatial data. In *SPIE Visualization and Data Analysis*, Jan. 2011.
[17] J. R. Miller. Attribute blocks: Visualizing multiple continuously defined attributes. *IEEE Computer Graphics & Applications*, 27(3):57–69, May/June 2007.
[18] G. Perlman. Data analysis programs for the UNIX operating system. *Behavior Research Methods and Instrumentation*, 12(5):554–558, 1980.
[19] Y. Tang, H. Qu, Y. Wu, and H. Zhou. Natural textures for weather data visualization. In *Tenth International Conference on Information Visualization*, pages 741–750, July 2006.
[20] T. Urness, V. Interrante, E. Longmire, I. Marusic, S. O'Neill, and T. W. Jones. Strategies for the visualization of multiple 2D vector fields. *IEEE Computer Graphics & Applications*, 26(4):74–82, 2006.
[21] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2nd edition, 2004.
[22] C. Weigle, W. Emigh, G. Liu, R. M. Taylor II, J. T. Enns, and C. G. Healey. Effectively visualizing multi-valued flow data using color and texture. In *Graphics Interface*, pages 153–162, 2000.
[23] E. Williams. Experimental designs balanced for the estimation of residual effects of treatments. *Australian Journal of Scientific Research, Series A*, 2:149–168, 1949.