An Approach to the Perceptual Optimization of Complex Visualizations

Donald H. House, *Member, IEEE Computer Society*, Alethea S. Bair, Colin Ware

Abstract – This paper proposes a new experimental framework within which evidence regarding the perceptual characteristics of a visualization method can be collected, and describes how this evidence can be explored to discover principles and insights to guide the design of perceptually near-optimal visualizations. We make the case that each of the current approaches for evaluating visualizations is limited in what it can tell us about optimal tuning and visual design. We go on to argue that our new approach is better suited to optimizing the kinds of complex visual displays that are commonly created in visualization. Our method uses human-in-the-loop experiments to selectively search through the parameter space of a visualization method, generating large databases of rated visualization solutions. Data mining is then used to extract results from the database, ranging from highly specific exemplar visualizations for a particular data set, to more broadly applicable guidelines for visualization design. We illustrate our approach using a recent study of optimal texturing for layered surfaces viewed in stereo and in motion. We show that a genetic algorithm is a valuable way of guiding the human-in-the-loop search through visualization parameter space. We also demonstrate several useful data mining methods including clustering, principal component analysis, neural networks, and statistical comparisons of functions of parameters.

Index Terms—Data mining, Evaluation/methodology, Theory and methods, Visualization techniques and methodologies.

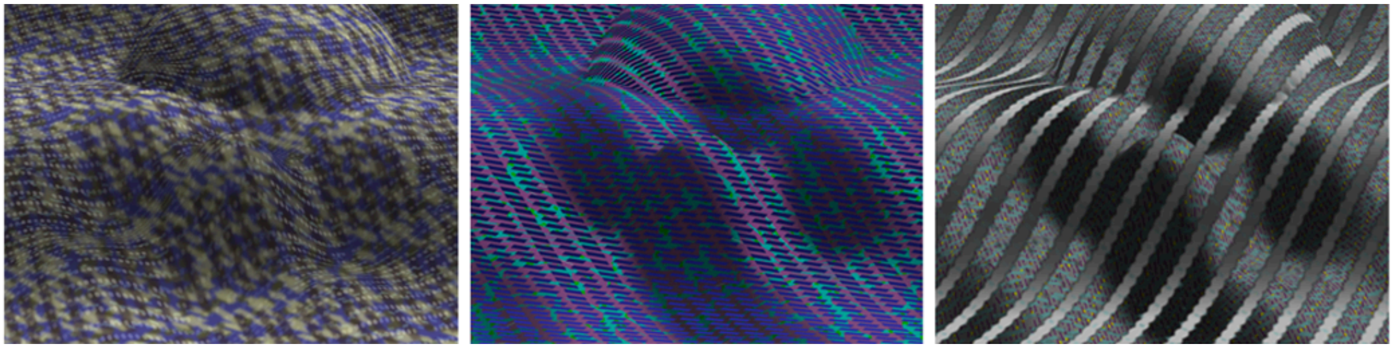


Fig. 1. Experimentally determined equally good solutions to layered surface texturing problem. Solutions are highly diverse. From House et al. [11].

Introduction

This paper is a contribution to the growing literature on evaluation and design in scientific and data visualization. It proposes a new way in which experimental evidence regarding the perceptual characteristics of a visualization method can be collected, and how this evidence can be explored to discover principles and insights to guide visualization design. The paper elaborates a theme that we began in [11].

Optimizing the presentation of a visualization is difficult, because for any complex configuration to be visualized, and any specific method to do the visualization, there can be many equally good solutions. For example the three images in Figure 1, although radically different from each other, have all been experimentally shown to be equally strong solutions to the problem of texturing overlapping surfaces for simultaneous viewing in a stereoscopic display. Further, evaluation criteria can range from the objective to the subjective, and from the absolute to the relative. We can ask questions that are as concrete as "Can the user efficiently perform task *A* using this visualization?", as vague as "Is this visualization visually pleasing?", or that are simply comparative such as "Is this visualization an improvement over earlier ones?". A powerful approach to visualization optimization must be designed to be able to accommodate these ambiguities and divergences. Our feeling is that optimization methods in current use in visualization all have serious shortcomings in the face of the complex problems that we are often asked to tackle.

Optimization is clearly built on evaluation. In visualization, the most prevalent evaluation method is that of expert evaluation by the experimenters and their colleagues. This method is at the heart and soul of our field, driving its development, as it is highly suited to the innovation of new algorithmic methods. As suited as this method is to sustaining methodological progress, it is clearly colored by individual biases of the experimenters. A recent trend, especially in Information Visualization, has been the employment of user studies (see for example [4, 20, 28]).

These often take the form of usability and case studies of particular visualization tools [22]. This approach is less prone to individual biases, but any conclusions on visualization optimality are highly confounded by the software engineering and user interface characteristics of the tools. Finally, a more novel approach to optimizing visualizations has been to engage highly skilled illustrators in the visualization design phase [16], so that their artistic expertise can be brought to bear on the problem. Although the visualization results obtained in this way are often highly intriguing, it is difficult to extract general principles from such studies, as results are highly variable. For the reasons enumerated, our position is that none of the above approaches to evaluation are ideal for uncovering the fundamental insights necessary to dependably design for optimality.

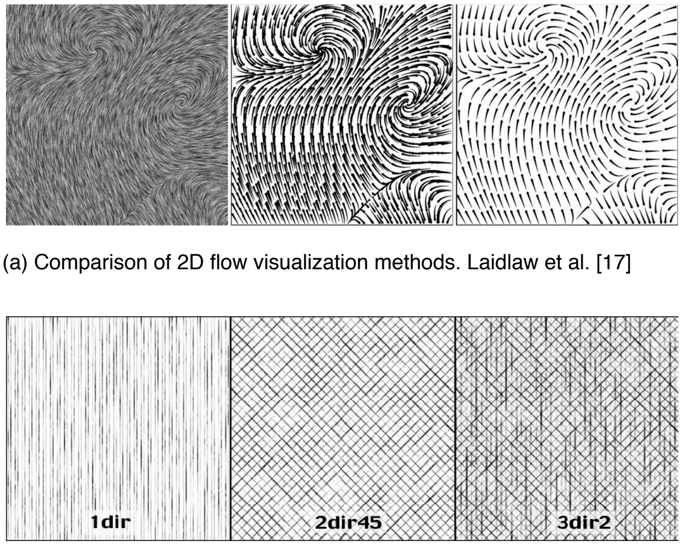
We cannot talk about evaluation without first identifying the fundamental insights that are required to guide design. Clearly, any useful theory must be built on these fundamental insights. For example, the rediscovery of the constructive rules for perspective drawing in the Renaissance was founded on the insight that by projecting rays from points in a scene through a single view-point in space, and intersecting these rays with a fixed viewing plane, we uniquely determine the correct location of these points in the perspective drawing. All rules for perspective construction can be derived from this one insight.

The idea that insight is necessary for the development of theory is reasonably obvious, but a more subtle point is that even evaluation requires these insights. Doing evaluation without a sound theoretical basis can lead to very wrong conclusions. For example, the famous Tacoma Narrows Bridge collapse of 1940, shown in Figure 2, occurred because designers failed to consider how winds could excite the torsional natural frequencies of the structure [27]. All of the necessary static structural calculations and tests were made to evaluate the soundness of the bridge. However, because the designers did not consider the dynamic characteristics of the bridge, they failed to do the necessary evaluations. It took a wind of only 40 m.p.h. to produce huge oscillations of the roadbed and eventual structural failure.

Fig. 2. 1940 Tacoma Narrows Bridge collapse. From [27].

To get at fundamental insights, perceptual and cognitive psychologists have a long history of using controlled experiments to conduct psychophysical studies in areas relevant to visualization. For example, there have been several parametric studies of the way we perceive curved surfaces [3, 5, 21, 23, 24, 29]. The idea of a controlled experiment is that we control all variables that might account for variance in the experimental outcome, keeping most fixed while allowing only one or two to vary. Statistical analysis of experimental results then seeks to look for correlations between variance in experimental outcomes, and changes in the variables that were allowed to vary.

There have been notable attempts within the visualization community to adapt these methods to evaluating visualization techniques. For example, groups lead by Laidlaw [17], and Interrante [15] have used this approach, allowing the enumeration of the strong and weak points of a number of 2D flow visualization techniques (Figure 3a), and theoretical ideas regarding which texture features enhance identification of the shapes of surfaces (Figure 3b).



(a) Comparison of 2D flow visualization methods. Laidlaw et al. [17]

(b) Texture patterns for visualizing surface shape. From Kim et al. [15].

Fig. 3. Examples from controlled studies in visualization.

However, even controlled experiments are suspect when addressing the kinds of complex problems we often try to address in visualization. The problem is that controlled experiments start with the premise that when we fix certain parameters their perceptual effects also remain fixed. But, this ignores possible visual interactions among parameters. For example, the well known color interaction phenomenon, expounded by the artist Josef Albers [1], leads one to see the central colored rectangle on the left in Figure 4 as having the same color as the field on the right, and the central rectangle on the right as having the same color as the field on the left. This is just an illusion, since both central colors are actually identical, matching the color of the strip across the bottom. Even a cursory study of the great variety of visual illusions will lead one to discover a number of other examples of such perceptual interactions, that arise when the visual system is attempting to make sense of imagery. Once we admit visual interaction among parameters, we are faced with the daunting task of doing an exhaustive study of all parameter settings in order to truly understand the perceptual effects of these parameters.

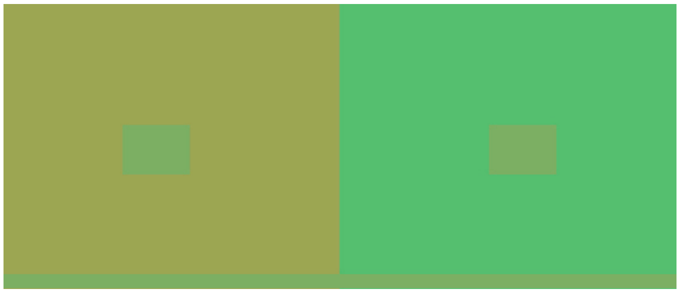


Figure 4 – Perceptual interaction of colors. Although they appear quite different, the two central squares are the same color as the strip across the bottom.

Exhaustive studies of parameters are impractical for all but the simplest of problems. To illustrate this point, suppose there are only 10 parameters relevant to optimizing a particular visualization and each is given a modest 5 levels. The result is then 5^{10} conditions. Supposing that each requires 20 settings by a subject to obtain a reliable estimate of error, then the result is a need for nearly 200 million separate measurements for a full parametric study. If each measurement took 5 seconds it would take almost a billion seconds to carry out the experiment. This is over 30 years per experimental subject, and would clearly test the patience of even our most dedicated graduate students!

Thus, our conclusion is that all of the accepted methods for evaluating or designing visualizations have either built-in biases or practical limitations that limit their efficacy in laying either firm theoretical foundations or practical guidelines for the reliable design of high-quality visualizations. The remainder of this paper is devoted to describing the conceptual framework for a new alternative methodology, and providing an example of how this framework was utilized to organize a study of texturing for overlapping surface visualization.

METHODOLOGY

We call our method for exploring the perceptual characteristics of a visualization method the *human-in-the-loop* approach. It consists of two phases. The first is the *experimental phase*, where subjects are engaged in a data gathering process that has some of the aspects of a controlled experiment, while much more rapidly exploring the entire space of visualization parameters. The experimental phase is followed by the *data-mining phase*, where statistical methods are used to discover perceptual information or explore hypotheses using information gathered in the first phase.

Experimental phase

As diagrammed in Figure 5, in broad outline the experimental phase of our method consists of the following steps:

1) choosing a class of visualization problem to investigate,
2) choosing a visualization method for this class of problem,
3) developing a parameterization of the method suitable to the problem, so that a vector of parameters, together with the chosen method, controls the visual representation,
4) running a set of experiments designed to search the parameter space, guided by user evaluations,
5) building a database of the rated visualization solutions visited during the experiments.

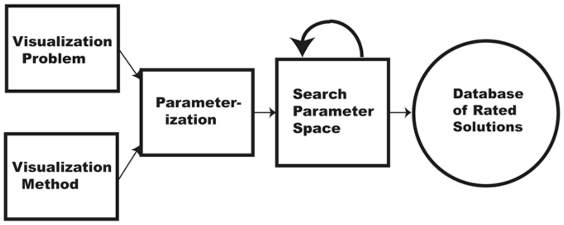


Figure 5 – Experimental phase of human-in-the-loop method

Step 4, parameter space search, is at the heart of our method and is where the name human-in-the-loop is derived. This search engages human subjects (who can be either expert or naïve) to rate solutions on grounds that the experimenter considers appropriate to the problem. Ratings can range from highly subjective: such as asking subjects to provide a rating based on their opinion of the quality of a visualization, to highly objective: such as measuring subject performance on a perceptual task involving the visualization. Whatever rating method is chosen, in most cases, it makes sense for the algorithm used to search the parameter space to use these ratings to guide the search so that high quality areas of the parameter space are most finely searched, and poor quality areas are sparsely searched.

Although a number of ways could be used to do the parameter space search, we have found that a genetic algorithm [7], using genes corresponding to the visualization parameters, is well suited to this task. This follows work by Dawkins [6], Sims [26] and Greenfield [8] who coupled image generation with user feedback in the context of a genetic algorithm, to control aesthetic content of images. It also follows the notable study by He et al. [9] who used this approach to optimize parameters controlling the transfer function used in rendering volume data sets.

We use a genetic algorithm operating over a relatively small population or generation. We begin by generating a population of random genomes. Since each genome encodes all of the visualization parameters, it defines a complete visualization. Thus, it is straightforward to produce the visualization for each member of the population and to employ a human subject to evaluate these visualizations. When all of the members of a generation have been evaluated, the genomes, together with their ratings, are written to an experimental database. The ratings are then used to control probability of breeding to produce the next generation. Genomes encoding high quality visualizations have a higher probability of “mating” and producing offspring. These offspring share genetic material (i.e. visualization parameters) from both parents, and constitute the next generation, which is in turn subjected to evaluation. After breeding, a small percentage of the genes are randomly mutated to help avoid converging too quickly on a local minimum in the parameter space. This procedure is iterated for as many generations as necessary to produce a homogenous population, as judged subjectively by the subject. In a few hours or days, the process is capable of methodically producing a large database of evaluated visualization solutions that samples the visualization parameter space most densely in regions of high quality, and least densely in regions of low quality.

Data mining phase

The experimental phase leaves us with a database of rated solutions sampling the visualization parameter space, to be used in the data-mining phase. The goal of this phase is to glean information about what leads to effective visualizations and what detracts. Following the curved arrow in Figure 6 we go from least to most general possible results. Even a single study should enable us to identify a set of exemplary visualizations for a specific problem. Having identified strong specific solutions, it is highly useful to identify parameter groupings that could be varied together without degrading visualization quality, providing a mechanism for variation about a specific solution. Further, it would be useful to be able to specify sets of default parameter settings for visualization applications, or better, design guidelines that that can be followed by designers of visualization solutions or software. The ultimate object would be to gain enough of an understanding of the structure of the visualization parameter space to guide the construction and testing of perceptual theory. We have experimented with a number of data-mining methods for extracting information from the database that span most of this range of outcomes.

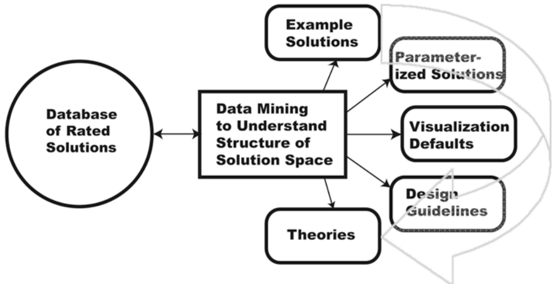


Figure 6 - Data-mining phase of human-in-the-loop exploration

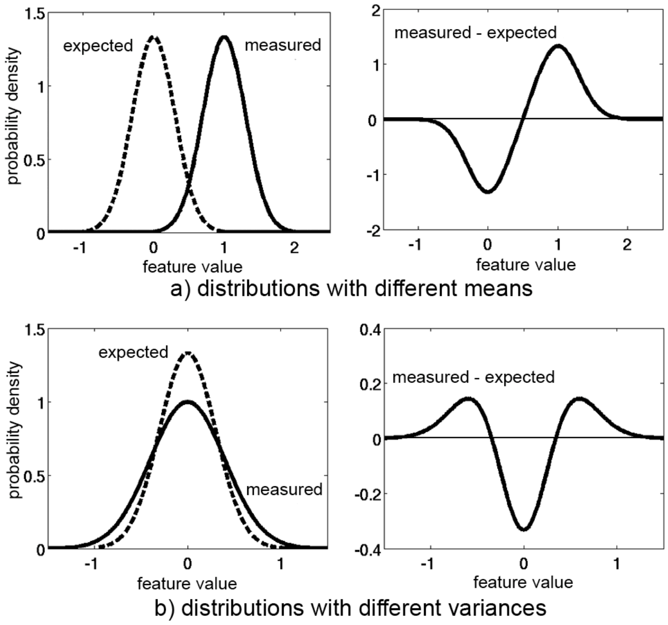
The most straightforward data mining approach is clustering of highly rated visualizations [18], with cluster membership determined by Euclidian distance in parameter space. Each cluster then consists of a number of solutions that share high ratings, and can be represented by an exemplar taken from the center of the cluster. These exemplars can then be used as the example solutions indicated in Figure 6, and the clusters from which they are taken constitute constellations of solutions around these exemplars that can be used to provide variety.

Principal component analysis is an excellent way to move from example solutions, obtained from the clusters, to parameterized solutions - allowing methodical variation of parameters without degrading visualization quality. Since, in our approach, a vector of parameters encodes each visualization, it is straightforward to find the principal components of a cluster of highly rated solutions. These principal components are vectors in parameter space aligned in the direction of maximum parameter variance within the cluster. If we pick a cluster center as the origin of a coordinate system in parameter space, then the principal components give coordinate directions along which we can expect to vary visualization parameters without degrading the quality of the visualization. Thus, they could be used in an application to allow the user to vary a default visualization while maintaining visual quality.

To more broadly capture the global structure of the data in the database, we have used neural network analysis [10]. We train a network on the parameters in the database to be able to produce ratings as outputs. The trained neural network then gives us a black-box function producing ratings from parameters. Given such a network, ideally we could examine the structure of connections in the network to infer relationships among parameters and ratings. However, since the units of a neural network use nonlinear transfer functions, and the number of connection pathways from inputs to outputs is large, simple examination of network connectivity to deduce such relationships is difficult. However, a linear connectivity analysis can produce some tentative guidelines, which can be further tested in other ways.

Although none of these data-mining methods can lead directly to easily generalizable conclusions, each method can contribute to insights about which parameter combinations lead to strong visualizations and which detract. These insights can then be further explored, with the goal of developing concrete guidelines or even theories. Visual inspection of the exemplars found by cluster analysis can lead to descriptions of their salient features, which gives an idea of the variety of parameter setting strategies that lead to good visualizations. Examination of interrelationships across clusters might also help to broaden these strategies. Examining the parameter-space vectors constituting the principal components of clusters can uncover which directions one can move in parameter space without detracting from visualization quality, and which directions tend to strongly affect quality. Within a trained neural network, examination of patterns of strong connectivity from inputs to outputs can lead to hypotheses about parameters or sets of parameters that appear to strongly affect visualization quality. When results derived from several data mining methods all lead to similar hypotheses, these become good candidates for closer inspection by finer grained statistical approaches.

The tentative hypotheses about combinations of parameters suggested by the other analyses can be reinterpreted as nonlinear functions of these parameters. One approach to testing such a hypothesis is to express it as a function of parameters, and then examine the function's distribution. For example, the distribution of the function for highly rated data can be compared to the expected distribution assuming the function has no effect on visualization quality (obtained by generating a random population). Figure 7 illustrates this method of analysis. The left-hand column shows hypothetical example distributions that are Gaussian in shape, where the measured distribution differs from the expected. In Figure 7a the mean of the experimental data is shifted to the right, in 7b the means are the same but the measured data exhibits a higher spread. The plots to the right show the difference between the measured distribution and the expected distribution. These can be read to conclude that in order to create good visualizations, a function value of 1 is preferable to 0 in 7a. Values to either side of 0 are preferable to values near 0 in 7b. Because of the large bias towards 0 in the expected distribution, the fact that the measured data had many values near 0 does not imply that 0 is the best value. Since there are many ways for parameters to interact, on average there will be a tendency for good distributions to look like random distributions. So, what we must search for are significant deviations.



a) distributions with different means

b) distributions with different variances

Figure 7 – Measured distributions vs. expected distributions.

We have found kernel density estimates to be useful for constructing distributions of continuous functions of parameters, and histogramming for discrete cases. Under the null hypothesis that the function of parameters being considered has no effect, the expected distribution can be found by assuming a random distribution for every parameter in the function. The measured distribution is simply the distribution of the particular function of parameters on highly rated visualizations. The significance of the difference between the two distributions can be measured using the Kolmogorov-Smirnov 2-sample test [19], or using confidence intervals on bins of the data.

Layered Surface Texturing Example

In order to illustrate our perceptual optimization method more clearly, we offer the following example of a pilot study of texture mapping to enhance the visualization of layered surfaces. Details of preliminary forms of this study are presented in [2, 12]

One of the most important yet perceptually difficult problems in data visualization is that of displaying one surface overlaying another. Just a few of the numerous applications of layered surface visualization include: medical imaging – to see the shape of different tissues overlying each other; geological applications – to see how geological layers are situated with respect to each other; and oceanography – to see how surfaces defined by temperature changes relate to the underlying seabed topography.

What makes layered surface visualization such a difficult perceptual problem is the combination of occlusion and visual confounding between the images of the two surfaces. For example, shape-from-shading information can be impossible to perceptually separate. Further, while making the top surface highly transparent can reduce occlusion, this surface then becomes difficult to see. When encountering layered surfaces in the real world – like viewing a scene through a screen of shrubbery – we have the advantage of highly coupled vergence and accommodation cues. As our eyes bring one surface into binocular registration and sharp focus, the other surface becomes blurred, thus reducing the confounding of visual cues. On a computer screen, however, both surfaces are presented on the plane of the display surface, so that we lose the advantage of the three-dimensionality of the real problem. Ware and Frank [30] have shown that we can help matters by providing some depth information through both stereoscopic viewing and motion parallax. Further, Interrante et al. [13] found that adding distinct partially-transparent textures to the layered surfaces can help to distinguish them. Nevertheless, even with all of these perceptual aids, there is still a strong tendency to visual confusion.

The visualization problem

We defined the visualization problem to be: how to choose pairs of tiled, draped textures for two surfaces so that, when the surfaces are overlaid and viewed in stereo and in motion, they optimally reveal the shapes of both surfaces and do not perceptually interfere with each other. It has been shown that textures grown to conform to features of a surface are very powerful in conveying surface shape in monocularly viewed static images (see especially Kim et al. [15]). However, in our study we consider only simple tiled, draped nonconformal textures. The ubiquity of use of such textures in visualization applications argues for their continued study. Further, we are aware of no evidence that conformal textures improve on simple draped textures under stereo viewing. Because textures can be arbitrarily complex, this is not an easy problem to solve. It can take ten or twenty parameters to define a single complex texture with a reasonable set of texture elements and color components. Further, there is the issue of how the textures should be oriented with respect to the viewpoint and the surface topography. Due to the number of parameters, it is difficult to see how much progress can be made on this problem simply using controlled studies.

The visualization method

The method that we chose for studying the layered surface problem was to fix viewing and surface parameters, while varying the textures applied to the two surfaces. This has elements of a controlled study, but the experimental variables are extremely complex. The scene consists of the overlay of the two surfaces shown in Figure 8. The bottom surface is a flat plane with hills in the center defined via a Gabor function. The top surface has a long-period sinusoidal wave whose front is nearly perpendicular to the viewing direction, and a large dome-like structure. The planes of the two surfaces are parallel, tilted away from the camera by 30 degrees and separated by slightly more than the height of the tallest feature on the bottom surface (to avoid interpenetration of the surfaces). The scene is lit using a single parallel light source with direction vector <1, 1, 1>, and shading is done using a simple 70% lambertian + 30% ambient shader, without specular highlights. The surfaces are viewed in stereo using a frame sequential CRT display and shutter glasses. The scene is rocked about the center vertical screen axis to provide motion parallax so that both stereo and motion cues are available to resolve depth.

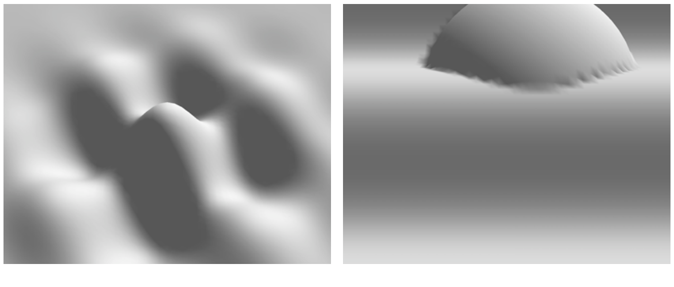


Figure 8 - Surfaces used in the layered surface experiment

Problem parameterization

We developed a parameterized texture space that would allow us to test a number of texture attributes that we felt might bear on the layered surface problem. For example, it can create a superset of the textures used by Watanabe and Cavanagh [32]. The overall texture attributes that we decided to parameterize were: 1) orientation of the texture on the surface, 2) foreground transparency, 3) density of pattern, 4) regularity of pattern (i.e. structured vs. random), 5) softness (i.e. soft vs. hard edges), and 6) background color. The attributes of individual texture elements making up the pattern were: 1) transparency, 2) size, 3) linearity (i.e. long vs. short strokes vs. dots), 4) orientation, and 5) color. Texture tiles were algorithmically constructed from the parameters by first building a background layer and then drawing in three separate sets of texture elements. The background layer is of a constant HSVα color, and the HSVα colored texture elements consist of one set of dots and two sets of linear strokes.

We build our textures from layers of elements, each grown on a lattice. Figure 9 demonstrates how the various parameters affect texture features drawn within the lattice cells. Figure 9a is a standard set of lines on a 4x4 lattice. Line length and width parameters can be varied to change the line size and aspect ratio, as shown in Figure 9b. The number of rows and columns in the lattice can be varied to create large-scale ordering of the features, like the 20x4 lattice shown in Figure 9c, where vertical lines are perceived, although the actual feature lines are horizontal. Features are given a rotational offset between -90° and 90° (45° shown in Figure 9d). Features are randomized in several ways: rotational jitter is shown in Figure 9e, translational jitter in Figure 9f, (horizontal and vertical jitter are separate parameters). Figure 9g demonstrates the result when the drawing probability parameter, which is the probability that a feature is drawn at each lattice cell, is set at 0.5. Figure 9h demonstrates blurring, which is controlled by a parameter that adjusts Gaussian low-pass filter width. Dots, as shown in Figure 9i, use the same parameters as lines, except that dots use the width parameter as a diameter, and ignore length and rotational parameters.

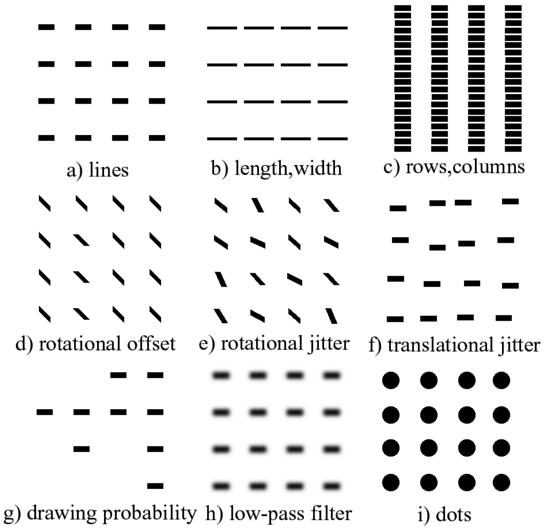


Figure 9 – Effects of parameters used to control texture features

A single texture tile is composed of a background layer and three lattice layers, one of dots and two of lines. Using this structure we parameterize a single texture by a vector of 61 elements, so that 122 parameters describe a texture pair. Seven parameters per texture determine overall appearance, and each of the three lattice layers requires 18 parameters. Complete textures can vary across a range from a fully transparent background with opaque texture elements (giving the illusion of texture elements floating in space) to a translucent background with translucent texture elements (giving the illusion of a continuous textured surface). Figure 10 shows two different foreground/background texture pairs generated in this manner. The foreground textures have transparencies, so for illustration purposes they are shown composited over orange.

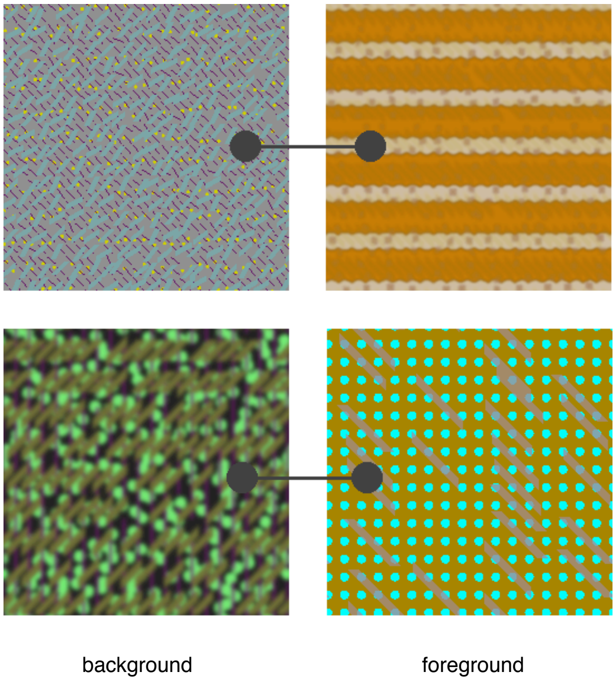


Figure 10 - Example texture tiles

Parameter space search

The experimental trials, used for human-in-the-loop visualization parameter space search, were controlled by a genetic algorithm. Trials were conducted in the following way. For each presentation, a subject was shown the surfaces from Figure 8 layered over the top of each other, and textured according to a trial set of parameters. They were asked to qualitatively evaluate their ability to clearly see the shapes of both the bottom and top surfaces. To make sure that subjects understood what they were to be looking for, at the start of each session the subject was shown the layered surfaces with hand-designed textures that did a reasonable (although not optimal) job, of showing both surfaces. The rating scale was 0-9, and input was made using a single numeric key press on a standard keyboard. These ratings were recorded with each texture pair, and were used to determine breeding fitness in the genetic algorithm. For the genetic algorithm, a single generation consisted of 40 texture pairs. Each of these pairs was presented in sequence until all were evaluated. Once a full generation was evaluated, breeding between textures was done using a two-point crossover approach, with the probability of a texture pair being selected for breeding determined by the experimental ratings. For our pilot study we used five subjects, each completing three full experimental trials of about 15 generations. Subjects were all familiar with computer graphics applications, so in that sense they could be considered experts. A trial was brought to an end when the textures in a generation were deemed to be fairly homogeneous. To reduce the effect of fatigue, subjects were able to save results at the end of any generation, and continue again at a later time. One complete experimental trial took about three working hours. Each trial successfully converged to produce a generation with a high percentage of the textures receiving high ratings. Figure 11 shows two image snapshots, with different texture pairs on the bottom and top surfaces, taken from two different points in our experimental trials.

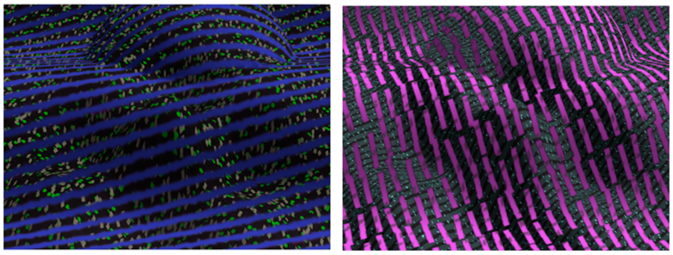


Figure 11 - Example presentations in layered surface experiment

Experimental database

From all of our experiments we obtained a database of 9720 evaluated surface texture pairs. In this database about 5% of the texture pairs have very low ratings (0 or 1), while about 34% have very high ratings (8 or 9). Figure 12 provides a comparison of the expected distribution of ratings given a completely random data set (dashed line – constructed from the ratings of the first generations only) vs. the distribution of ratings obtained over all trials (solid lines). It is clear that the algorithm focuses most of its time on exploring fruitful areas of the parameter space. Overall our process generated solutions at a rate of approximately 130 per hour with good solutions being produced at a rate of one every 2.5 minutes.

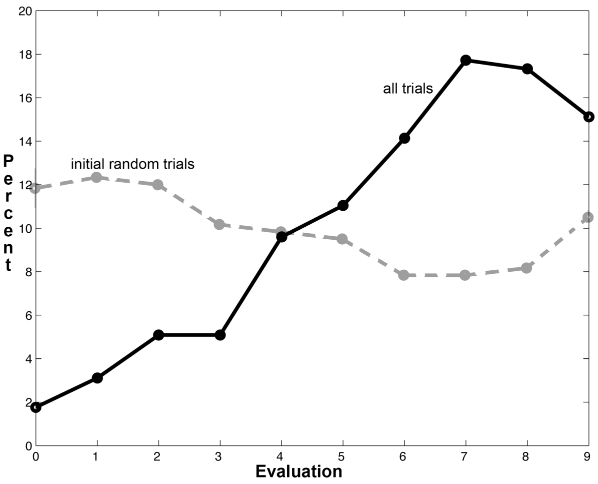


Figure 12 - Ratings in experimental database vs. random set

Data mining

We have experimented with a number of methods for gleaning information from our experimental database. These are surveyed below.

cluster analysis

Our earliest data-mining attempt was using cluster analysis on highly rated visualizations. Figure 1 (on the first page) shows cluster medians from three of the many clusters found in the database. Clusters were formed using a hierarchical-nearest-neighbors approach [14], so they are not typically spherical but can have elongated shapes in parameter space. What is immediately apparent from examination of Figure 1 is that these clusters are quite diverse in their structure and appearance. We found that in most, the percentage opacity of the top surface texture was a key factor, with texture elements being fully opaque and the background being fully transparent, as in the center and right image in Figure 1. However, there were several good solutions having a milky translucent surface scattered with small texture elements, as in the left image in Figure 1. Other indicators are that many, but not all, of the good solutions have texture components that differ greatly in size between the foreground and the background. Less obvious, but still apparent is a tendency for more structure on the top surface and a more random appearance on the bottom surface.

principal component analysis

We use principal component analysis of clusters of highly rated solutions to discover directions in parameter space along which visualizations may be varied without disturbing the quality of the visualization. The middle image in Figure 13 shows the median texture in a cluster. To its left and right are visualizations generated by following the first principle component in both directions from the cluster mean. The changes in orientation, color, and texture granularity do not degrade the visualization. The features represented by the principal component vectors can therefore be considered free parameters when constructing good textures. Unfortunately, this method does not provide specific rules for making good textures, since variation across the parameter space is ignored. However, it does give an indication of which parameters are more important.

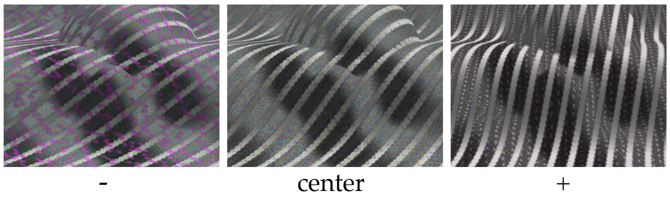


Figure 13 - Variants following first principal component

In analyzing the principal components corresponding to the largest eigenvalues, several trends were apparent. Comparable parameters always varied more on the top surface than the bottom surface. Also, with the exception of transparency, the comparable parameters on the surface background varied more than those for the texture elements. This implies that careful choice of settings for bottom surface characteristics are more important than those for the top, and that texture feature characteristics are more important than the texture background. Hue and saturation variables had more variation than value, leading to the conclusion that certain values of the value parameter are likely to be much better than others while hue and saturation were less important. Interestingly, parameters encoding the shape of features, such as the number of rows and columns in the lattice, size and shape of the elements, and randomness of the features, always varied less than the color parameters. This indicates that features must have good placement, size and shape before parameters like color, rotation and filtering can have much of an effect on visualization quality.

neural network analysis

Neural network analysis allows us to move away from cluster centers and consider the global structure of the parameter space. We built a 2-layer back-propagation network, as shown in Figure 14. The 122 input nodes each correspond with one of the parameters. These are fully connected to 20 hidden units, which in turn are fully connected to 10 output units. Each output unit corresponds to one of the texture ratings. When a texture is input as a vector of parameters, the output is a classification (0-9). Using only 20 hidden units provides a large data reduction from the 122 inputs, but the network learned to categorize accurately when trained on a training set drawn from the database. Figure 15 shows the histogram of neural network outputs when all visualizations rated 9 in a test set are used as inputs. Although not experimentally determined, this histogram appears to be well within the range of variability of human judgment. Histograms for all of the other ratings are similar, with correct mean and low spread.

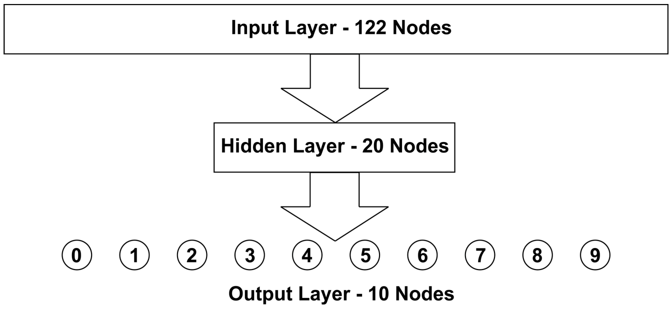


Figure 14 – Neural network structure. The network has one input node per parameter, and one output node per rating.

The non-linearity of the network units prevents a simple analysis of weight vectors. However simply looking at which features had large magnitude positive or negative weights leading to an output node proved interesting. Examining the weights to output unit 9 (most highly rated) supports the following hypotheses. The top surface transparency should be high, little low-pass filtering should be done, and rotation of the overall texture from horizontal should be high. Widely separated small lines seem to be preferred, with little horizontal or rotational jitter but large vertical jitter. Interestingly, only a single set of lines is indicated on the top surface, and the use of dots is not strongly indicated. This corresponds with the indication from clustering that there should be more structure on the top surface. On the bottom surface, high background value seemed preferred. In contrast to the top surface, the use of small, randomly placed dots with high value and saturation was indicated.

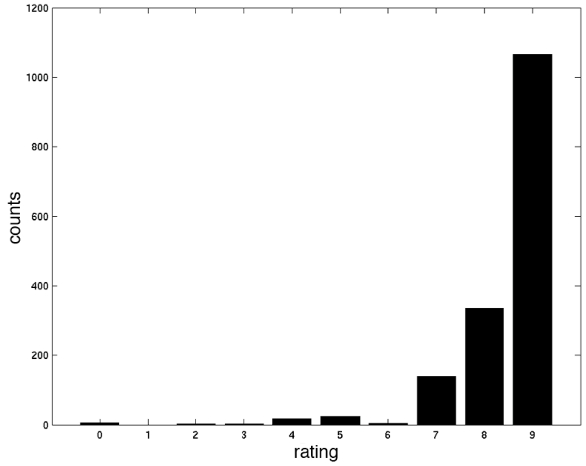


Figure 15 –Distribution of neural net scores for visualizations rated 9.

Hypothesis testing

In order to explore the tentative hypotheses indicated by the above analyses, we used comparisons of parameter function distributions. The null hypothesis distributions were generated as described previously, using randomly generated parameter sets. The measured distributions were given by the textures in the dataset rated either 8 or 9. There were 3087 such textures. We used Matlab [19] for the analysis, and to produce kernel density plots for visual inspection. The distributions were intentionally over-smoothed to minimize any small distribution biases that could arise simply from the genetic algorithm search method. As a measure of how different the measured distribution is from the null hypothesis, the Kolmogorov-Smirnov two-sample test was performed. This produces a p-value, which is the probability that the two distributions are the same. Usually any p less than 0.05 is enough to reject the null hypothesis and declare the distributions different. Also, as an estimate of which peaks are significant, 95% confidence intervals were constructed on bins of data with the same width as the smoothing kernel.

Both the principle component analysis and the neural network analysis suggested that parameters affecting texture feature shape were highly important for creating good textures. Two of these shape parameters are feature aspect ratio and grid cell aspect ratio. For our analysis we define the aspect ratio to be the smaller of the two lengths divided by the bigger of the two lengths. Thus, the scale goes from near 0, meaning long and thin to 1 meaning square. Features with a low aspect ratio will tend to create strokes similar to hatching along a surface. Grid cells with a low aspect ratio (i.e. that are elongated) will tend to globally align the features creating long lines across a surface.

Figure 16 (top) shows the difference between the measured and expected distributions for the feature aspect ratio for the top and bottom surface separately. Error bars indicate 95% confidence intervals. The very low p values indicate that the curves are significantly different from random. Aspect ratios near 1 and 0.25 were preferred for the top surface. Overall, slightly squarer features were preferred than for the random distribution. On the bottom surface, however, more elongated aspect ratios were preferred overall. These results beg the question of whether differences in aspect ratio between the top and bottom surface features affect visualization quality. Figure 16 (bottom) shows the clear result that a difference in feature aspect ratio is helpful and the top surface should have squarer features. The fact that the curve is generally below the axis for aspect ratio differences below zero, and above for differences above zero means that for highly rated textures the expectation is greater than chance that aspect ratios on the top are larger than on the bottom, and less than chance that they are greater on the bottom.

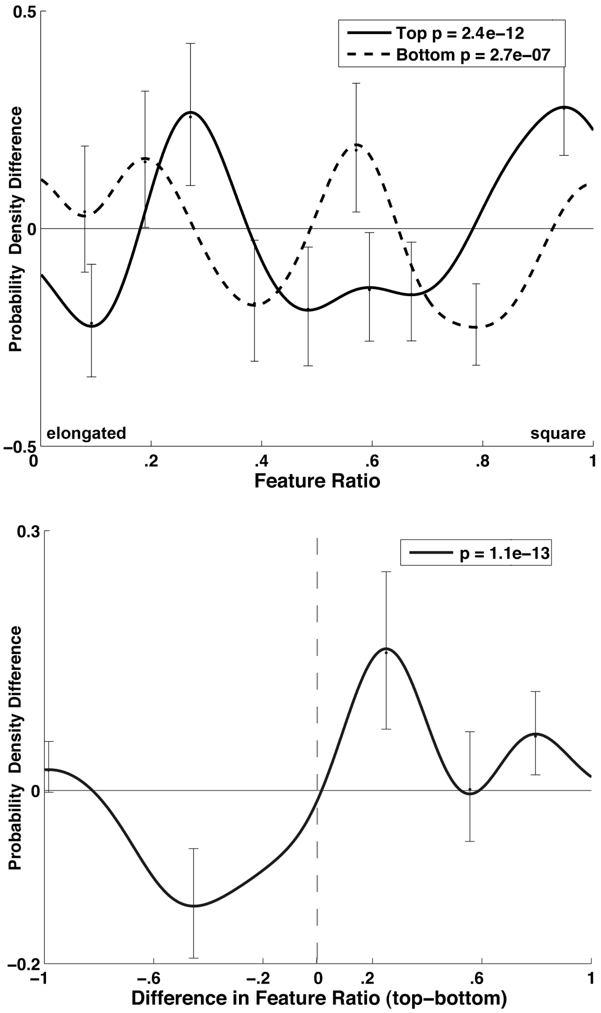


Fig. 16. Feature aspect ratio distributions.

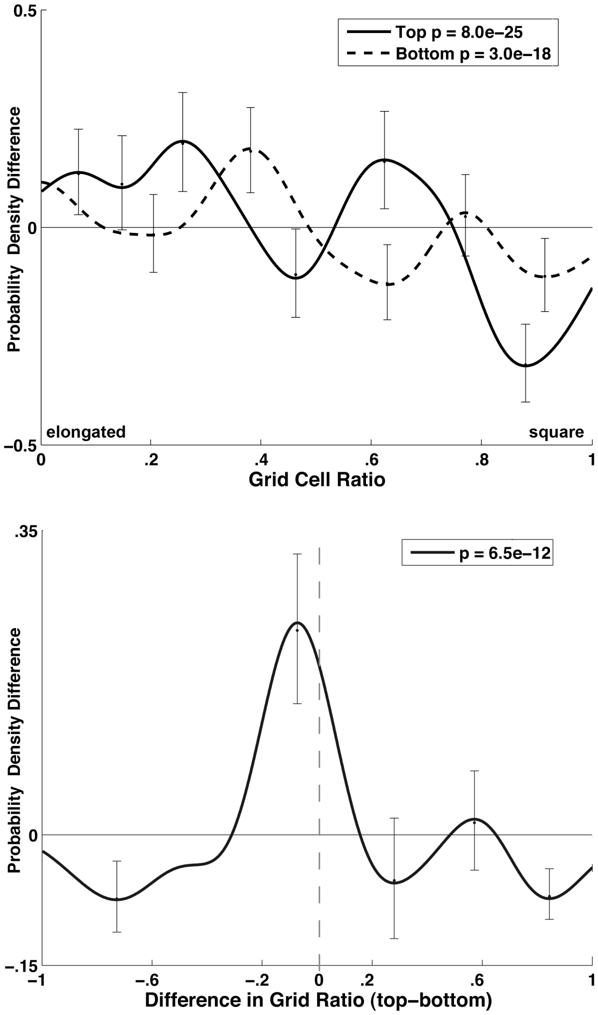


Fig. 17. Grid aspect ratio distributions.

Analyses of grid aspect ratios are shown in Figure 17. The separate analysis of top and bottom surfaces in Figure 17 (top) shows a preference for both surfaces to have anisotropic (i.e. non-square) grids, with the preference more apparent for the top surface. This is consistent with the large proportion of texture cluster centers (see Fig. 1) that had dramatic, large-scale lines on the top surface. The way in which the preference peaks of the two distributions tend not to overlap suggests that a difference in grid aspect ratio might also be helpful for minimizing confounding of the two surfaces. Interestingly enough, Figure 17 (bottom) shows a clear preference for the top and bottom surfaces to have the same grid aspect ratio. (Note that this says nothing about the size or orientation of the grid).

For some of our hypotheses it proved easier to use image analysis than to work with functions of the parameters. One example is the average opacity of the top surface. Figure 18 shows the difference between the distribution of average opacity of the highly-rated textures and the distribution of opacities from randomly-generated textures. It shows a preference for coverage near either 20% or 50%, and an expected avoidance of 0% and 100% coverage.

Lastly, the distributions of the average color value, plotted in Figure 19, show a strong preference for the bottom surface having a value near either 0.5 or 0.8 of full scale. The top surface distribution of values on the other hand, was not significantly different from the random distribution according to the Kolmogorov-Smirnov two-sample test.

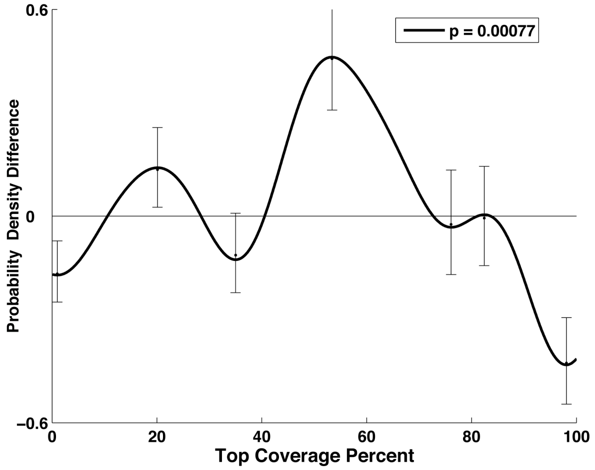


Fig. 18. Top surface coverage distribution.

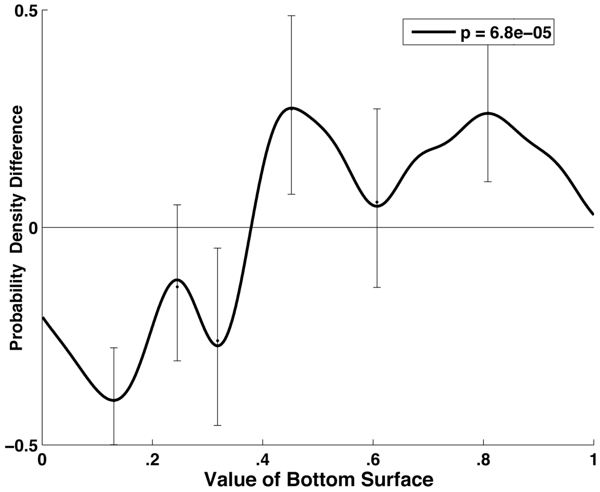


Fig. 19. Bottom surface color value distribution.

Summary of layered surface results

Based on our full range of tests, including many not shown here, we were able to develop a set of general guidelines for smoothly varying layered surfaces:

- Textures across the two surfaces should have a relative rotation of at least 20^0.
- Coverage (net opacity) on the top surface should be between 20% and 60%.
- Features on the top surface should be larger in scale than on the bottom surface.
- The top surface should appear more structured and the bottom surface more random.
- Texture features should be more square on the top surface and more elongated on the bottom surface.
- Color values on the bottom surface should be near either 50% or 80% of full-scale brightness.
- Color saturation should be higher on the bottom surface than on the top.
- Color hues can be chosen freely.

A set of crossed-eye stereo pairs of textures constructed following and then ignoring these guidelines is displayed in Figure 20. These give some idea of the actual experimental presentation, although their low resolution and the absence of motion cues make them considerably weaker than they appear on screen. Figure 20a shows images of three different texture pairs with parameters hand-selected according to our guidelines. The top two images differ in that the top image uses two layers of texture elements for the top surface, while only one layer is used for the middle image. Although not quite as effective as the top image, the middle image still shows both surfaces clearly. The bottom image shows that dramatically changing hues has little effect on the ability to clearly see both surfaces. In Figure 20b we break various rules. The top image shows the result when the top surface has a very fine grid similar to the bottom surface. The fine texture on the top blends with the texture on the bottom and it becomes very difficult to see the shape of the top. For the middle image we increased randomness on the top surface, making the top shape harder to pick out. For the bottom image, we made the top surface value brighter than the bottom, which makes the top surface very easy to read but the bottom surface is now difficult to see.

Discussion

Many of the guidelines we discovered applying to the problem of two-surface visualization are, to our knowledge, entirely novel. If these findings stand the test of scrutiny by more rigorous methodologies, they may eventually become the basis for new perceptual theories that apply to the perception of transparent surfaces. We must add that it is possible that our generation method may have biased some of the results and further testing will be needed before they can be regarded as more than tentative.

It is clear that a simple genetic algorithm approach to searching the visualization parameter space is not ideal. For the individual experimental subjects it is a slow process to arrive at consistently strong visualizations. One of the frustrations, especially for an expert subject, is the inability to use domain knowledge in the search process. In the layered texture experiments, we would frequently see a texture that could be improved immensely in obvious ways, but all we could do was to score it and move on. We have two ideas to augment the GA approach, that we feel will help. The first is to implement an “islanding” capability [25] that allows creation of an “island” population of textures, all nearby in parameter space to a texture that the subject finds interesting. The island population could then be evolved by itself or later merged back into the general population. Our second augmentation to the GA would be to provide an interface that would allow direct “tweaking” of the visualization parameters. Tweaked visualizations could then be inserted into the population to affect future evolution.

Although neural network analysis cannot give us a mapping from ratings to parameters, it can be helpful in a number of ways. Most especially, it might be used to numerically estimate gradients in parameter space. We are looking at ways in which this could be coupled back to the data-gathering phase to assist in guiding the search through the visualization parameter space. Thus, results of previous experiments could be used to make new experiments more efficient. The neural network can also be used to more densely populate the database by randomly generating parameter sets, scoring them using the network, and using them to selectively fill in gaps.

One aspect of our experimental methodology that we have not yet investigated is inter- and intra-subject variability in the evaluation of visualizations. Our informal observation is that subjects learned the surface shapes quite quickly; so learning effects beyond the first generation of 40 textures were minimal. However, it is more likely that the scaling of ratings continued to accommodate to the subject's experience. For this "proof of concept" study, we considered this to be a minor effect, but this needs further exploration. Examining intra-subject variability could be easily incorporated into the existing genetic algorithm, by reinserting randomly chosen presentations into the experiment during the scoring process. Inter-subject variability could be examined by similar methods but making sure that all subjects see and rate some small percentage of identical presentations. Alternatively or additionally, we could run a set of later trials in which we ask all of our subjects to rescore the same representative set of presentations.

We are currently at work incorporating what we have learned from our preliminary study of layered surface texturing into a new study. This will use a greatly reduced parameter space (24 vs. 122 parameters), surfaces with fixed but broad spatial frequency content that vary in shape with each trial, and a somewhat more objective evaluation criterion – ease of finding a fixed number of features of varying scale. This experiment will use two or three evaluation criteria per presentation: ability to find various sized protrusions on each of the two surfaces, and overall aesthetic quality. To determine the overall fitness rating for the genetic algorithm, we plan to use a product of the two surface bump counts, to which the weighted aesthetic score will be added. For the datamining, however, these scores can be analyzed separately. Varying the surfaces with each presentation, and providing a wide range of spatial frequencies in the shapes of the surfaces, will remove the bias toward very smooth surfaces inherent in the current experiment. Finally, the experiment will be carried out on a custom designed high-resolution Wheatstone stereoscope [31], providing visual resolution matching the human eye.

Conclusion

We have argued for and demonstrated a new human-in-the-loop approach to the optimization and evaluation of visualization solutions. Our demonstration problem led us to discover previously unknown guidelines for a complex visualization problem. The method is capable of accounting for the perceptual interrelationships that occur among parameters in complex visualizations. It can also account for the mix of objective and subjective factors affecting the quality of visualizations. We have described a practical way of selectively sampling from the space of solutions to a visualization problem and experimentally evaluating these samples. We have also proposed a number of data-mining techniques for extracting useful information from the database produced during this process.

A distinct advantage of our method, compared with controlled experiments, is that the database we produce contains information about all of the dimensions in the parameter space. Collecting this data is exacting and time consuming, but not significantly more so than for a controlled experiment, which typically gives us information about only one or two dimensions. Furthermore, the experimental database can be shared with the general community, and can continue to be a useful information source for discovering and testing new hypotheses well beyond the time of the experiment.

Ultimately, our goal is to arrive at solidly grounded theory regarding perceptual issues affecting visualization quality. Although we have been able to use our method to develop solid guidelines, these are not yet grounded in theory. However, we believe that our approach can be an important element in developing and testing theory. While our approach is described as a two-phase process, the technique can also feed back on itself, using results from data mining or theoretical hypotheses from other forms of analysis to form hypotheses that can be tested by further experimentation. Thus, a series of experiments can be generated to further test and refine results. For example, new experiments could start with a hypothesis, gleaned from the data mining, about how particular parameter settings affect visualization quality, and, starting with exemplar solutions also from the data mining, test this hypothesis by methodically varying these parameters from their base values in the exemplars.

We feel that the most fruitful direction for enhancing the power of this new methodology will be in developing more powerful data mining techniques. There is a wealth of related literature in far-flung disciplines such as psychology, sociology, economics, and ecology. We feel that exploration of this literature in the context of visualization holds promise of greatly contributing to development of the field.

Acknowledgement

This work was supported in part by the National Science Foundation ITR's 0326194 and 0324899, the Center for Coastal and Ocean Mapping – University of New Hampshire, the Visualization Laboratory – Texas A&M University, and the Texas A&M Department of Architecture.

References

[1] J. Albers, *Interaction of Color*, Yale University Press, 1963.

[2] A. Bair, D. House, C. Ware,. "Perceptually Optimizing Textures for Layered Surfaces," *Proceedings of Symposium on Applied Perception in Graphics and Visualization*, pp 67-74, 2005.

[3] M.J. Black, R. Rosenholtz 1995. "Robust Estimation of Multiple Surface Shapes from Occluded Textures" *International Symposium on Computer Vision*, pp. 485-490, 1995.

[4] C. Chen, M.P. Czerwinski, "Empirical Evaluation of Information Visualizations: An Introduction," *Int. J. Human-Computer Studies*, 53, pp. 631-635, 2000.

[5] B.G. Cumming, E.B. Johnston, A.J. Parker, "Effects of Different Texture Cues on Curved Surfaces Viewed Stereoscopically," *Vision Research*, 33(56): pp. 827-838, 1993.

[6] R. Dawkins, *The Blind Watchmaker*, Harlow Logman, 1986.

[7] D.B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 2nd edition, 1999.

[8] G. Greenfield, "Color Dependent Computational Aesthetics for Evolving Expressions," *Bridges: Mathematical Connections in Art, Music, and Science; Conference Proceedings*, pp. 9-16, 2002.

[9] T. He, L. Hong, A. Kaufman, H. Pfister, "Generation of Transfer Functions with Stochastic Search Techniques," *Proceedings of IEEE Visualization 96*, pp. 227-234, 1996.

[10] S. Haykin, *Neural Networks, A Comprehensive Foundation, Second Edition*, Prentice-Hall, Upper Saddle River, NJ, 1999.

[11] D. House, A. Bair, C. Ware. "On the Optimization of Visualizations of Complex Phenomena," *Proceedings of IEEE Visualization 2005*, pp. 87-94, 2005.

[12] D. House, C. Ware, "A Method for the Perceptual Optimization of Complex Visualizations," *Proceedings of Advanced Visual Interfaces*, pp. 148-155, 2002.

[13] V. Interrante, H. Fuchs, S.M. Pizer, "Conveying Shape of Smoothly Curving Transparent Surfaces Via Texture," *IEEE Trans. on Visualization and Computer Graphics*, 3(2): pp. 98-117, 1997.

[14] S.C. Johnson, "Hierarchical Clustering Schemes," *Psychometrika*, 2, pp. 241-254, 1967.

[15] S. Kim, H. Hagh-Shenas, V. Interrante, "Conveying Shape With Texture: Experimental Investigations of the Texture's Effects on Shape Categorization Judgments," *IEEE Trans. on Visualization and Computer Graphics*, 10(4): pp. 471-483, 2004.

[16] D.F. Keefe, D.B. Karelitz, E.L. Vote, D.H. Laidlaw, "Artistic Collaboration in Designing VR Visualizations," *IEEE Computer Graphics and Applications* (Pending publication).

[17] D.H. Laidlaw, M. Kirby, C. Jackson, J.S. Davidson, T. Miller, M. DaSilva, W. Warren, M. Tarr, "Comparing 2D Vector Field Visualization Methods: A User Study," *IEEE Trans. on Visualization and Computer Graphics*, 11(1): pp. 59-70, 2005.

[18] J. Marks, B. Andalman, P.A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, "Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation," *Computer Graphics (Proceedings of SIGGRAPH 97)*, pp. 389-400, 1997.

[19] The MathWorks Inc., *Matlab 6.5.1.199709 Release 13*, 2003.

[20] E. Morse, M. Lewis, K.A. Olsen, "Evaluating Visualizations: Using a Taxonomic Guide," *International Journal of Human-Computer Studies*, 53 (5): pp. 637-662, 2000.

[21] J.F. Norman, J.T. Todd, F. Phillips, "The Perception of Surface Orientation From Multiple Sources of Optical Information," *Perception and Psychophysics*, 57(5), pp. 629-636, 1995.

[22] C. Plaisant, "The Challenge of Information Visualization Evaluation," *Proceedings of Advanced Visual Interfaces*, pp. 109-116, 2004.

[23] V. Ramachandran, "Perceived Shape from Shading," *Scientific American*, August, pp. 76-780, 1988.

[24] B. Rogers, R. Cagnello, "Disparity Curvature and the Perception of Three-Dimensional Surfaces," *Nature* 339, May, pp. 137-139, 1989.

[25] C. Ryan, "Niche and Species Formation in Genetic Algorithms," in Lance Chambers, Ed., *Practical Handbook of Genetic Algorithms*, Vol. 1, CRC Press, Inc., Boca Raton, pp. 58-73, 1995.

[26] K. Sims, "Artificial Evolution for Computer Graphics," *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25, pp. 319-328, 1991.

[27] D.A. Smith, "Case Study and Analysis of the Tacoma Narrows Bridge Failure," *99.497 Engineering Project*, Department of Mechanical Engineering, Carleton University, Ottawa, Canada, March 29, 1974.

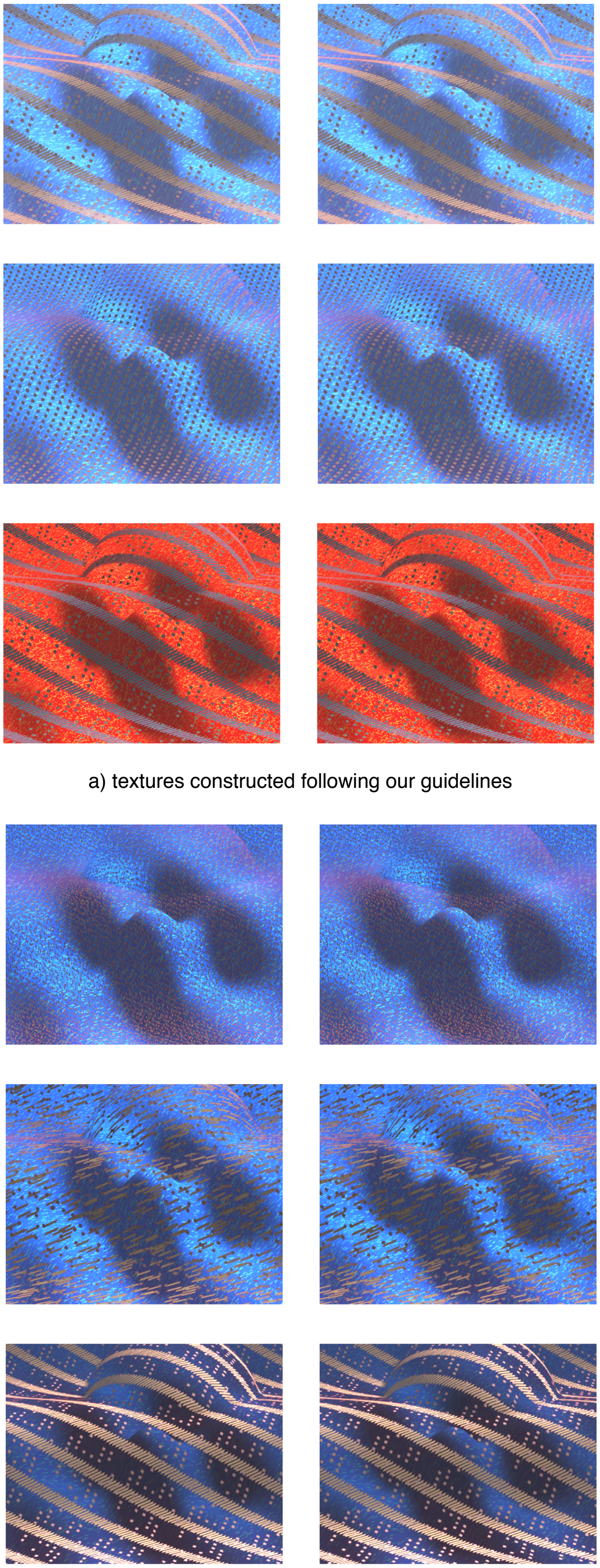
[28] J. Stasko, R. Catrambone, M. Guzdial, K. McDonald, "An Evaluation of Space-Filling Information Visualizations for Depicting Hierarchical Structures," *International Journal of Human-Computer Studies*, 53 (5): pp. 663-694, 2000.

[29] J.T. Todd, R. Akerstrom, "Perception of Three-Dimensional Form From Patterns of Optical Texture," *J. Experimental Psychology: Human Perception and Performance*, 13 (2): pp. 242-255, 1987.

[30] C. Ware, G. Frank, "Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions," *ACM Transactions on Graphics* 15(2): pp. 121-140, 1996.

[31] C. Ware, P. Mitchell, "Reevaluating Stereo and Motion Cues for Visualizing Graphs in Three Dimensions," *Proceedings of Symposium on Applied Perception in Graphics and Visualization*, 2005.

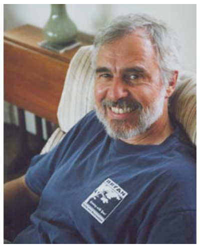
[32] T. Watanabe, P. Cavanagh, "Texture Laciness," Perception, 25, pp. 293-303, 1996.



a) textures constructed following our guidelines

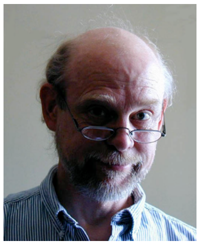
b) textures constructed violating our guidelines

Fig. 20. Stereo pairs of textured layered surfaces



Donald House is a Professor in the Department of Architecture at Texas A&M University, where he guided the curricular development of a unique, interdisciplinary Master of Science program in Visualization Sciences. In computer graphics he is best known for the development, with David Breen, of interacting-particle methods for the simulation and visual display of woven cloth. Theirs and the work of a number of others is collected in their volume *Cloth Modeling and Animation*, published in 2000. Most recently he has been working on novel methods for perceptual optimization, capable of dealing with the complexity of real visualization tasks. He holds a Ph.D. in Computer and Information Science from the University of Massachusetts, Amherst, an M.S. in Electrical Engineering from Rensselaer, and a B.S. in Mathematics from Union College.

Alethea Bair is a Ph.D. student at Texas A&M University in the Department of Architecture, where she is pursuing a program of study in Visualization Sciences. She received a degree in physics from the University of Illinois, Urbana/Champaign, and as an undergraduate did summer research applying visualization techniques in Physics at the University of Wisconsin. Her research interests include perceptual optimization, data mining and pattern analysis, and image analysis.



Colin Ware is Director of the Data Visualization Research Lab, which is part of the Center for Coastal and Ocean Mapping at the University of New Hampshire. He is cross appointed between the Departments of Ocean Engineering and Computer Science. Ware specializes in advanced data visualization, especially in applications of visualization to Ocean Mapping. He combines interests in both basic and applied research, holding advanced degrees in both computer science (MMath, Waterloo) and in the psychology of perception (Ph.D., Toronto). Many of his over 90 scientific articles relate to the use of color, texture, motion and 3D displays in information visualization. The 2nd edition of Ware's seminal book *Information Visualization Perception for Design* was published in 2004. He directed the initial development of the *NestedVision3D* system for visualizing very large networks of information, and has been instrumental in the creation of two visualization companies based on his research. He is currently leading a group developing *GeoZui3D*, which stands for GEO-referenced Zooming User Interface 3D.