Multi-layer Visualization: A Review of Selected Methods

Caesar Ogole, Julius Kidubuka
Institute for Mathematics and Computing Science
University of Groningen, The Netherlands
{C.Ogole, J.Kidubuka}@student.rug.nl

Abstract
While the advances in scientific visualization have made it possible to convert contextual data sets into conspicuous meaningful images, some areas still need further exploration. One of these challenges, which is the focus of this review, is the problem posed by the question: "Given large, complex and multi-dimensional data sets that represent overlapping surfaces and fields in the real world, what visualization technique can be applied to optimize the display of this class of images?" This problem is particularly difficult owing to the fact that solution methods to multi-layer visualization problems do not only involve many variables such as textures, colors, orientation and (the degree of) transparency of overlaying surfaces, but also, have to integrate user-centered issues such as user feedback. Human perception is core to the considerations. Moreover, the integration of these factors into a typical visualization system takes the form of parametizations of complex and highly interactive algorithmic procedures. No standard guidelines to select suitable sets of parameters exist. In this paper, we review three different techniques of enhancing multi-layer visualization.

1 Introduction

The goal of scientific visualization is to transform sequences of numbers and character strings into images from which useful information can be inferred. These sequences fall into categories, each of which may possibly be representing some attribute(s) of the entity in question. The representative values are called datasets. Since most entities are complex, it is only possible for them to be represented by large and multi-dimensional datasets, which in general, have many different data elements. The complexity of the datasets poses a big problem to visualization processes, particularly, in the case where the encoded image information in the data patterns is multi-layer in nature. This problem is not merely a classic case of investigation in theoretical computations but it is also an area having vast applications in the real world, for example, in medical imaging where, in practice, a subject has to view the tissues overlain by skeleton or vice-versa.

The problem of multi-layer visualization can be understood quite easily by thinking of an image scenario where there are a collection of N surfaces (or fields) that overlap one another in space (Figure 1). *How do we view the overlaying surfaces*? Suppose the top surface is more opaque, can we still view the bottom surface conspicuously without having its shape distorted in any way? Obviously, the viewer will not be able to extract useful information. This is likely to render the visual system, and hence the applied technique, not user-friendly or even useless.

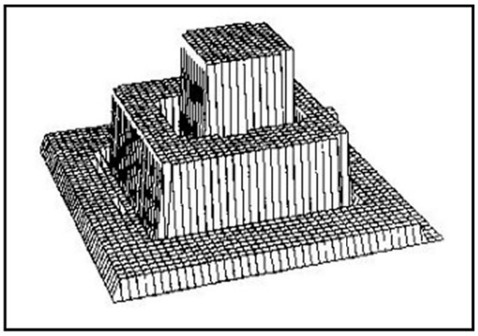

Fig.1. The "wedding cake" illustration of overlapping surfaces

Thus, this paper provides a brief review of some of the existing methods of enhancing multi-layer visualization. At some point during its use, the technique employed in such a system requires that a user chooses values for system parameters so as to guide the system in improving or refining its output. This is the actual transformation phase where the representative datasets are converted into multi-layer (overlapping surface) images. Variables of special interest include, but are not limited to, surface texture colors, orientation, opacity (or selective blur), shapes, distribution, sizes and segmentation. While it would be a good idea to study the effect of each of these variables over the others (one at a time), it is practically not possible given that the number of variables is very large.

2 Related work

The quest in finding effective solution to the problem of multi-layer visualization is not a new thing. Several attempts had hitherto been made by various researchers and it is these contributions that served as a starting point for the techniques reviewed in this paper. We give a brief overview of related work.

Methods of analyzing image textures using statistical techniques have been shown to work well under certain conditions [1], although it focused mainly on a single task (texture). The big setback with the previous methods was that they did not take into account other relevant image attributes. The texton theory [2], a contribution by Juliész, pointed out that early vision detects three types of texture features, namely, elongated blobs with specific visual properties (for example, colors and orientation), ends of line segments and crossings of line segments. Closely related to multi-layer visualization applications, Interrante[3] uncovered that if one or both surfaces are given spatially transparent texture, this can help to define and distinguish them. Interrante further reported that additional depth information provided through stereoscopic viewing and motion parallax can make them stand apart. However, none of these studies gave guidelines for choosing texture pairs that both optimally reveal the surface shape and do not interfere with one another. The problem of perception thus crops up.

All the works related to multi-layer visualization are about image surface texture, because surfaces in nature are generally textured. As explained by Gibson [4], texture is an essential property of a surface. A non-textured surface, he said, is merely a patch of light.

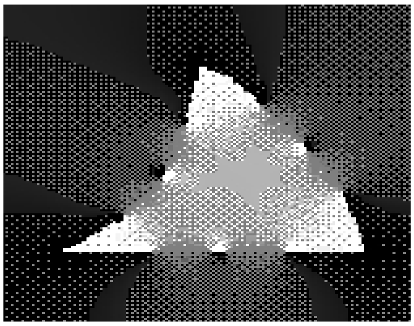

Fig.2. Textures as an important surface property

Some authors such as Dawkins [5] and Sims [6] went as far as giving useful hints into methodological boundaries in generating solutions to visualization problems. Notably, it is observed that the structure of a genetic algorithm provides a convenient means for visualization and user feedback. This is the heuristic that has been used extensively in the visualization techniques reviewed in this paper. Genetic algorithms are particularly good for this class of computational problems because they allow prioritization of operations within regions that are believed to be “promising” in the context of generation of better or improved results.

3 Review of Methods

In this survey, it is observed that no single technique is best. To this effect, each technique is first described to some satiable length and level of detail before the pros and cons are shown. The hope is that detailed understanding of these attempts will lead to further refinement by other researchers who will venture into addressing any pitfalls in the methodological designs.

3.1 Method for perceptual Optimization of Complex Visualizations

In this first technique [7], the tasks involved in the solution method can be divided into three stages. The first problem is to represent the datasets using appropriate data structures. Arrays (or vectors) of data elements, whose sizes depend on the dimensionality of the datasets, are used for this purpose. The term *gene* is used to refer to the encoded parameter vector, tuned as an input to the transformation procedure in the next phase. The idea of encoding datasets at this stage is to define the search space within which all derived solutions must lie. In the second phase, gnome is fed into the conversion process which is actually a routine that falls under the family of genetic algorithms. Notable at this phase is the involvement of the user in supplying parameter values that reflect preferences and hence, guiding the algorithm during the iterative output refinement. The third and final phase is to characterize the results so as to select the palatable solution. This is done according to the clustering criterion.

Why a genetic algorithm? The search space can be overwhelmingly large for exhaustive search. A genetic algorithm is useful because the search proceeds only towards promising regions, and is somewhat resilient in avoiding poor local minima.

3.1.1 Data Structure

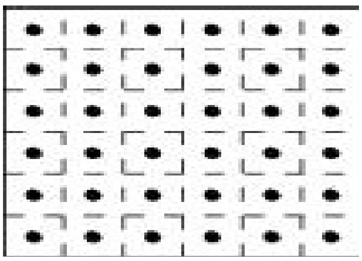

Fig 3. A 6X6 lattice with dots

In this representation, each texture tile is structured as a set of three lattices. Using standard texture mapping approach, a complete texture across a surface is tiled from a single base file. The tile is divided into a uniform square grid by the lattices.

3.1.2 The algorithm

The pseudo-code for the genetic algorithm is shown in Appendix 1. It is important to understand the terminology used in the procedure description. Each encoded parameter is referred to as a *gene.* Arrays of genes, or *genotypes,* are in turn stored in an array of *generation.* It is the generations (arrays of arrays, or multi-dimensional arrays) that are fed into the algorithm as inputs. The gene is usually encoded as an integer or floating point and each gene controls an aspect of a texture pair. Prior to the algorithm's first run, the generations are initialized to randomized values. Subsequent values depend on the user preferences that guide the interactive algorithm towards desired optimal generation values.

3.1.3 Results

On average, it was found that generation of acceptable solutions through the repetitive steps took two hours per subject. This process was successful in producing good results to the problem of visualizing overlaying surfaces for all the subjects. However, initial randomized data (generation) values for the texture pairs did not seem to have a good representation of the optimal solution values. Iteration run time decreased with increasing number of iterations, and after about twenty minutes, there seemed to be less distinction in the next and previous results. It was observed that opacity of the top surfaces greatly affected the visualization of overlapping surfaces (Figure 4). However, variation in colors apparently had negligible effects. This rendered colors only important in attribute display.

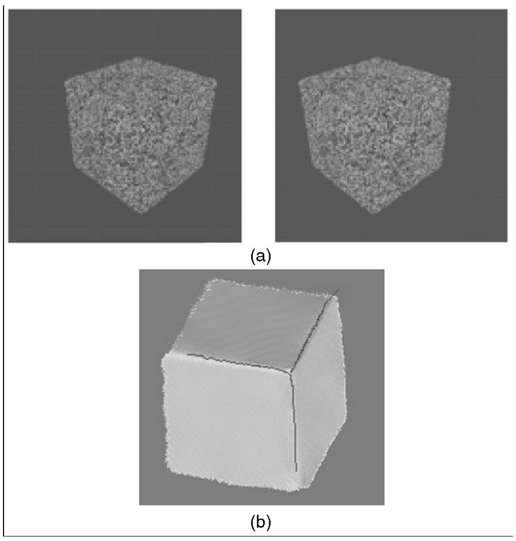

Fig.4. Degree of transparency affects visualizing overlapping surfaces. (Compare a and b)

3.2 Focus + Context Technique

Semantic Depth of Field (SDOF) [8] is the second technique employed in the display of overlaying surfaces. SDOF is based on the depth of field (DOF) effect borrowed from cinematography and photography that depicts objects (sharply or blurred) depending on their distance from the lens. Selective blur images based on relevance (rather than geometry) are used to guide the viewer's attention to the unblurred objects in the image. The aim of this technique is to efficiently and effortlessly present information to the user/viewer. The sub-techniques (stages) involved are split into three.

In the first stage, we make use of a process known as *preattentivity.* Two preattentive abilities are tested: being able to detect and locate a sharp object, and being able to estimate the percentage of targets among distractors. Experimental results showed preattentivity provides a reliable technique of finding sharp targets among blurred distractors. The accuracies for correct location of targets were very high (atleast 90 percent) or high (atleast 60 percent) depending on the blur level. Significant drop in accuracy was attributed to presence of the lowest blur level. Preattentive processes take place within a very short time (~200ms) and involves a limited set of features (such as orientation, closure, color, proximity, etc.) for which certain tasks (e.g. location, detection) can be performed with ease.

In the second phase, *interplay* is used. Figure 5 depicts an example of the images used for this part. The focus of interest is in the interaction of SDOF with other features (color and orientation are selected for use) because it is very likely that SDOF will not be used without any other visual cue apart from sharpness. Simple, disjunctive and conjunctive searches are tested as a way of detecting the presence of one or both features in the target.

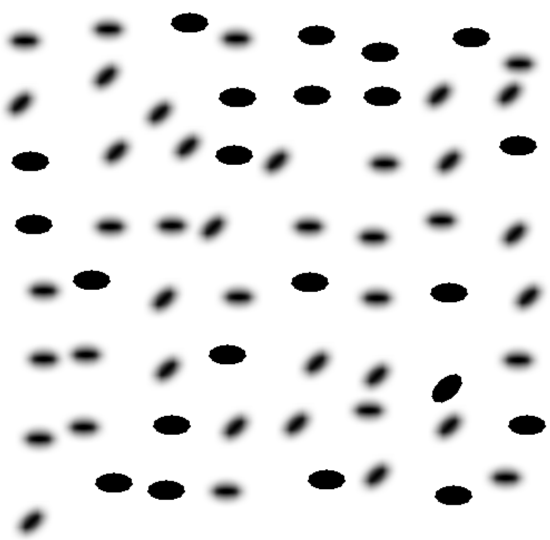

Fig.5. Example image for interplay

In the last stage (known as blur perception), the smallest difference that can be perceived in blur, and the rate at which "steps" in blur are perceived are assessed. This is based on the assumption that there is an exponential relationship between the blur level and the perceived blur. To do this, a test is performed (with the help of some participants) and it consists of a number of parts: testing the ability to tell whether or not two objects have the same blur level, the absolute thresholds of blur perception and finally to tell the perceived relation in blur in terms of a ratio of two numbers. Effectively, multilayer visualization is enhanced in that SDOF can then decide for every object whether to display it sharply or blurred (Figure 6). The decision is based on the object's current relevance.

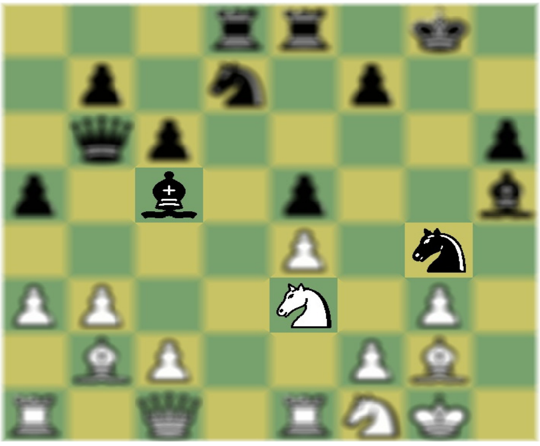

Fig. 6. Application example of a chess board, with the chessmen threatening the knight on e3 in focus (taken from Kosara et al.6)

3.2.1 Results

In terms of search time, SDOF was observed not to be significantly worse than color; this was perhaps the most interesting and surprising finding of the study. There was no significant difference between a simple search for colored or for sharp objects. The conjunctive searches for color and blur, orientation and blur, and color and orientation differed significantly from each other, with color and orientation being the slowest– each of these two features combined with SDOF was faster. Also, the conjunctive search for color and blur was not significantly slower than the simple and disjunctive searches, which was quite contrary to what was expected, because conjunctive searches usually were slower.

3.3 Local Value Estimation of Multiple Scalar Field using Oriented Sliver surfaces

To support the simultaneous display of multiple overlapping scalar fields, this texture generation technique [9] combines orientation and luminance that are selected based on psychophysical experiments that studied how the low-level human visual system perceives these visual features

3.3.1 Data Representation

Datasets in numerous practical applications can be viewed as a collection of *n* scalar fields that overlap spatially with one another. Rather than using *n* visual features to represent these fields, only two features are used: orientation and luminance. For each scalar field (representing attribute $\mathbf{A_i}$) a constant orientation $\mathbf{o_i}$ is selected; at various spatial locations where $a_i \in$ $\mathbf{A_i}$ value exists, a corresponding *sliver texture* is placed oriented at $\mathbf{o_i}$. The luminance of the sliver texture depends on a_i: the maximum $a_{max} \in$ $\mathbf{A_i}$ produces a white (full luminance) sliver, while the minimum $a_{min} \in$ $\mathbf{A_i}$ produces a black (zero luminance) sliver. A perceptually-balanced luminance scale running from black to white is used to select a luminance for an intermediate value. This scale was built to correct for the visual system's approximately logarithmic response to variations in luminance.

3.3.2 Procedure

Values in a given scalar field are given orientations (in degree angular measure, for example). Combining these orientations form sliver layers. Multiple scalar fields are displayed by compositing their sliver layers together.

With varying backgrounds orientation, say from 0° to 45°, in the intervals of 5°, (resulting in 10 different background subsections (0, 5, 10, ..., 45°), a discrete function *f(bg)* is defined for the different background orientations. The function *f* returns rotational differences in the intervals (*e.g.,d*= 5°, *d*= 10°, etc). Every possible target orientation was tested for each separate background. Several trials were run during the experiment

The goal of the experiment was to find how much counter clockwise rotation is needed to differentiate a group of target elements oriented $tg=bg+d_{ccw}$, *and* $tg=bg-dccw$ *where* bg *is* set of background elements orientations, d_{ccw} is the counter clockwise rotation.

3.3.3 Results

In general, using multi-factor analysis of variance (ANOVA) and least-squares line fitting, it was found out that target oriented $d = \pm 15°$or more from its background elements resulted in the highest accuracies and the fastest response times, regardless of background orientation.

4 Discussion

In this survey, we looked at three different methods applied in multi-layer visualization. It is observed that the major difference among the three lies in the steps followed in generating acceptable solutions to multi-layer visualization problem. Some of the procedures encompass more parameters than others

Whereas the first method (described in section 3.1) focuses mainly on selecting appropriate texture pairs that represent backgrounds and foregrounds of superimposing surfaces, the F+C technique (section 3.2) concentrates on utilizing the useful properties of Semantic Depth of Field (SDOF) for better visualization. The property used in SDOF is the fact that the selective blur aids in guiding user attention to the most relevant objects. On the other hand, the third visualization technique (section 3.3) is a texture generation method that combines orientation and luminance to support simultaneous display of multiple scalar fields.

As noted before, each method comes along with trade-offs. While the method for perceptual optimization of complex visualization has the power of handling multivariate characteristics of complex data, the criterion applied is not simple. It does not automatically produce solutions that are elegant. More abstractions are needed.

The method of local value estimation of multiple scalar fields, too, has a number of limitations, for example, as the number of attributes grows, it becomes difficult to find additional features to represent them. This technique does not provide a mechanism to handle interference (a phenomenon where different visual features will often interact with one another producing visual distortion). Nevertheless, it's more practical.

The SDOF approach (section 3.3) seems to be intermediate with respect to the first two techniques except that the optimality of its solution needs further investigation. Also, SDOF cannot be used as a full visualization dimension since in most cases it is used without any other visual cues.

From this survey, we may conclude the following.

Firstly, we observe that any approach to solving visualization problems, multi-layer in particular, requires interactivity between the user and the system so that perceptual problems can be solved through parameter adjustments, for example, by changing color, brightness etc. Analysis of the visualization techniques also reveals that genetic algorithms are a promising way of improving multi-layer visual problems. The solution to visualization problems cuts across various disciplines including computer science / graphics, psychology, photography, cinematography, etc. Finally, we see that appropriate combination of image attributes, such as textures, and orientations is important in visual display of overlapping surfaces. Poor combination of the values of these parameters results in orientations that may not be distinguished.

5 References

[1] REED, T. R. AND HANS DU BUF, J. M. A review of recent texture segmentation and feature extraction techniques.
Computer Vision, Graphics, and Image Processing: Image Understanding 57, 3 (1993), 359–372.

[2] JULÉSZ, B. A brief outline of the texton theory of human vision. Trends in Neuroscience 7, 2 (1984), 41–45.

[3] Interrante, V., Fuchs, H., and Pizer, S.M. (1997) Conveying shape of smoothly curving transparent surfaces via texture.
IEEE Trans. On Visualization and Computer Graphics 3(2)
98-117.

[4] Gibson, J.J. (1986) The ecological approach to visual perception. Lawrence Erlbaum Associates, Hillsdale, NJ.

[5] Dawkins, R. (1986) The Blind Watchmaker, Harlow
Logman.

[6] Sims, K. (1991) Artificial Evolution for Computer Graphics,
Computer Graphics 25, 319-328.

[7] House, C (2002) A method for the Perceptual Optimization of Complex Visualizations

[8] Kosara, R (2001)Useful Properties of Semantic Depth of Field for Better F+C Visualization

[9] Weigle, C (2000) Sliver Textures: A Technique for Local Value Estimation of Multiple Scalar Fields

[10] Christopher Healey and James Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, April 1999.

[11] Ivan Herman, Guy Melançon, and M. Scott Marshall. Graph visualization and navigation in information visualization:
A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, January-March 2000.

[12] SMITH, P. H. AND VAN ROSENDALE, J. *Data and visualization corridors report on the 1998 CVD workshop series.* Technical Report CACR-164 (sponsored by DOE and NSF), Center for Advanced Computing Research, California Institute of Technology, 1998.

Appendix 1: Pseudo-code for Perceptual Optimization of Complex Visualizations

```
G is current generation, V is next, N even
Generation G, V of size N;
Evaluation E of size N;
Phenotype P;
Restart from last saved or new random generation
if restarting from a previous session then
       (G,E)= LoadFromHistoryFile();
goto restart;
else
RandomlyInitialize(G)
endif
Main evaluate - breed - mutate loop
loop
extract visualization, display and evaluate
for each genotype Gi in G do
P = Phenotype(Gi);
Ei = UserEvaluation(Display(P));
endfor
SaveEvaluatedGenotypesToFile(G, E);
restart:
breed probabilistically based on evaluation, one
breeding pair produces 2 offspring
for k = 1 to N step 2 do breed
(i, j) = SelectBreedingPair(G, E);
(Vk, V k+1) = CrossoverBreed(Gi, Gj);
endfor
G = V; make new generation the current one
for each genotype Gi in G do
Mutate(Gi);    mutate    with    low probability
until UserRequestsExit; keep going until user quits
```

Acknowledgement

Special thanks go to Dr. Ronald van der Berg, Dr. R. Smedinga and Dr. J. Terlouw. We would also like to thank fellow student reviewers, Mr. Nicholas Edward Kirtley and Mr. Nestorgebruiker Dos Santos Pires for their contribution.

Expert Reviewer's Summary and Comment

Dr. Ronald van der Berg -- *An introduction is given to the problem of effectively visualizing several information layers simultaneously. Section two gives a brief overview of some existing methods to deal with this problem. Next, more detailed discussions of three more recent methods are presented. The paper concludes with a discussion section, in which the reviewed techniques are evaluated and compared to each other.*

In my view, this is a very reasonable paper and I could not think of any reason why not so accept it (for publication).