

# Compositing Color with Texture for Multi-Variate Visualization

Haleh. H. Shenas and Victoria Interrante

Department of Computer Science, University of Minnesota

## Abstract

Multivariate data visualization requires the development of effective techniques for simultaneously conveying multiple different data distributions over a common domain. Although it is easy to successfully use color to represent the value of a single variable at a given location, effectively using color to represent the values of multiple variables at the same point at the same time is a trickier business. In this paper, we provide a comprehensive overview of strategies for effectively combining color with texture to represent multiple values at a single spatial location, and present a new technique for automatically interweaving multiple colors through the structure of an acquired texture pattern.

**CR Categories and Subject Descriptors:** I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, Color, Shading, Shadowing, and Texture; I.3m [Computer Graphics]: Miscellaneous, Color Mixing, Perception.

**Additional Keywords:** color, texture, multivariate visualization.

## 1 Introduction

One of the important open research questions in the field of visualization is how to effectively convey information about multiple different data variables at the same location in a common domain, at the same time. The need for the effective visualization of such data arises all the time in fields such as statistics, geology, geography, environmental science, and a host of other application areas in which one seeks to develop an integrated understanding of multiple different data distributions in order to gain deeper insights into the nature of possible complex interactions between the different variables.

Color is perhaps the most ubiquitously employed visual variable for conveying information about the quantity of a particular data variable at a given point. Much effort has been devoted to the development of guidelines for the effective use of color in visualization [e.g. Rogowitz and Treinish 88] and the design of effective color scales [e.g. Brewer 94, Healey 96]. A variety of approaches have also been developed for effectively using color to simultaneously represent more than one data distribution [e.g. Robertson and O'Callaghan 86]; however all of these techniques are generally limited to the representation of no more than two or three different data values at a point.

In order to effectively convey the values of more than three different variables simultaneously at a point, it is necessary to augment the use of color with the use of texture [Levkowitz 91, Healey and Enns 99]. In this paper, we describe some new strategies for effectively using color with texture to represent multivariate data. We begin in section 2 by reviewing previous work on the use of (monochrome) texture for multivariate visualization. We also provide an overview of various strategies for combining color and texture in procedurally generated or hand-painted texture patterns. In section 3, we introduce a new technique for automatically integrating multiple different colors into the structure of an existing (acquired) texture pattern. In section 4 we show examples in which texture colorization is used for data visualization. Our ultimate goal in this paper is to show how to achieve a rich variety of different kinds of multi-colored patterns that can be used to accurately and intuitively convey up to six or more different variables simultaneously over the same spatial domain, thereby facilitating the potential discovery and deeper understanding of complicated inter-relationships between the multiple data variables.

## 2 Using Texture to Represent Multivariate Data

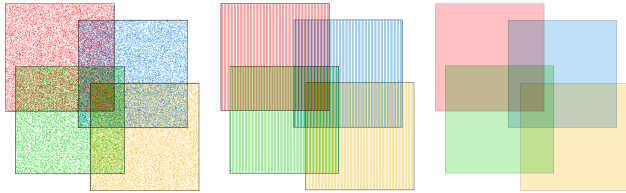
The first step in determining how to effectively use texture for multivariate visualization is to understand texture perception. A variety of texture metrics have been proposed. These include: contrast, coarseness and directionality [Tamura *et al.* 1978] repetitive vs. non-repetitive; low-complexity vs. high-complexity, and high contrast and non-directional vs. low contrast and directional [Rao and Lohse 1993a, 1993b] and periodicity, directionality and randomness [Liu and Picard 1996]. All identify three dimensions for texture perception, including a dimension pertaining to directionality. Thus we expect it to be possible to effectively represent at least four or more variables simultaneously via the combined use of color and texture.

Many researchers have developed methods for effectively using texture to convey information about multiple data parameters, particularly in the field of flow visualization. Examples of related works include: [van Wijk 91, Ware and Knight 92, Kiu and Banks 96, Healy and Enns 99, and Sanderson *et al.* 04].

The most common method for combining color and texture is color compositing, in which separate, individual color distributions are successively applied as semi-transparent color washes over an underlying texture pattern. Unfortunately, color compositing can be only successfully used to represent up to three different data distributions, as beyond that it becomes impossible to differentiate the individual color components that make up any particular composite color.

As an alternative to color compositing, Urness *et al.* [2003] introduced the concept of color *weaving*, in which an arbitrary number of different overlapping color distributions are displayed in a side-by-side manner. Gossett and Chen [2004] subsequently proposed a similar but more limited technique based on the use of Perlin noise textures.

The general technique of using a side-by-side color display rather than color mixing can of course be easily integrated into the synthesis of any procedurally generated texture pattern, through the definition of appropriate mappings between individual texture features and individual color distributions. Figure 1 shows a series of hand-constructed examples in which this ‘color weaving’ effect is demonstrated using two different procedurally-based texture patterns as a starting point, with the results of color compositing also shown for comparison.



**Figure 1:** Two different methods of integrating color into procedurally-defined texture patterns. Left: high frequency noise texture; Middle: stripe texture; Right: color compositing.

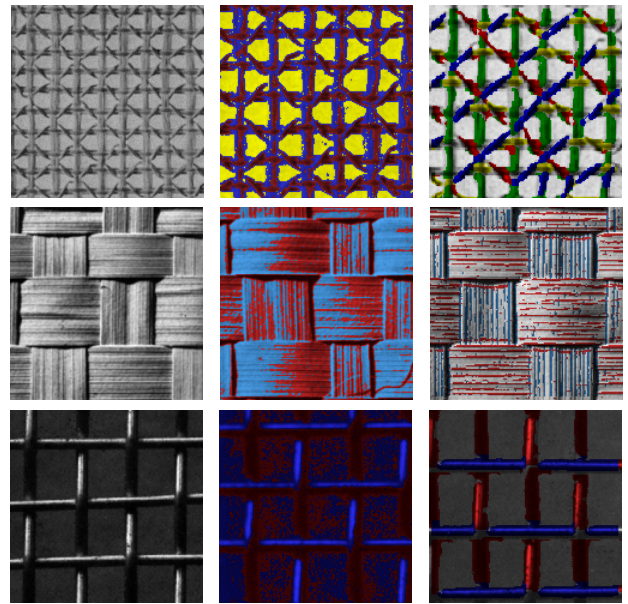
### 3 Combining Color with Arbitrary Textures

Although it is fairly straightforward to imagine a variety of methods for integrating the representation of multiple color distributions into a procedurally-defined texture pattern, it is less clear how, apart from resorting to hand-painting, we might effectively ‘weave’ the representation of multiple color distributions into an arbitrary acquired pattern. In this section, we describe two different approaches that can be used to automatically achieve this aim.

#### 3.1 Intensity-based Texture Segmentation

The simplest method for arbitrarily colorizing a monochrome texture pattern is to use the histogram of the texture pattern intensities to subdivide the texture image into a number of sections equal to the number of different colors, while ensuring that each section contains the same number of pixels. This method works best when the number of colors is small (our experience shows that using 2 or 3 colors generally produces satisfying results) and when the grayscale distribution of the texture has enough variety to allow for a reasonable segmentation. If  $n$  is the number of different colors we would like to represent over a texture swatch, we first divide the histogram of that texture image into  $n$  sections, each time stopping before the bin (we call this bin  $i$ ) that would bring the total number of pixels in the section to more than  $1/n$  of the total number of pixels. If the number of pixels at this division is exactly  $1/n$  we can record this point and continue dividing the rest of the histogram. Otherwise, we randomly select pixels from the bin  $i$  and add them to the current section until the total number of pixels in the section exactly reaches  $1/n$ . The remainder of the pixels belonging to bin  $i$  are added to the subsequent section. This process continues until  $n$  different sections have been defined and all pixels of the image have been assigned to a section.

In figure 2 we show three texture patterns for which this histogram division technique has worked fairly well. However, we note that significant important weakness of this approach is that the different regions thus defined will not necessarily represent an intuitive segmentation of the underlying pattern. This shortcoming will impede our ability to successfully use this method in future work to map data variables to perceptually meaningful texture dimensions, such as local pattern orientation.



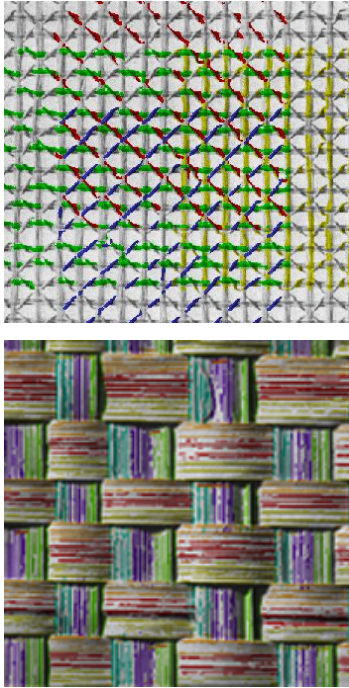
**Figure 2.** From top to bottom, patterns d052, d064 and d001 from the Brodatz album [1966]. Left column: the original grey scale images; Middle column: colorization based on the intensity histogram of the pattern; Right column: colorization applied using an orientation-based segmentation of the pattern.

#### 3.2 Texture Segmentation into Meaningful Blobs

An alternative approach, also shown in figure 2, is to decompose a texture pattern into multiple components based on the local orientation of directional elements of the pattern. For texture patterns containing foreground and background elements, we started by applying a region-growing algorithm to thresholded images of texture patches in order to differentiate the foreground and background elements in the patterns. For all texture patterns, we then applied a set of ‘steerable pyramid’ filters [Simoncelli and Freeman 1995] to each texture image. The steerable pyramid performs a polar-separable decomposition of the texture pattern in the frequency domain. This produces an independent representation of each of the sought-after scale and orientation components in the input image. To create the images shown in this paper, we used a pyramid that consisted of four different orientations ( $45^\circ$ ,  $90^\circ$ ,  $135^\circ$  and  $180^\circ$ ) at two different scales. As might be expected, we found that the filter responses for shaded natural textures presented a challenge for segmentation and filtering due to the inconsistent nature of the patterns. We found that the directional regions identified by the coarsest filters did not closely coincide with the structure of the texture within those regions, except for a small part close to the edges, while the directional regions identified by the finest filters did not exhibit strong continuity. In order to obtain a more desirable segmentation, we first combined the filter responses over multiple scales to obtain a stronger filter response for each direction with more accurate localization. We then used a region growing algorithm to find all the connected blobs in each of the desired directions. After deleting the blobs that were too small in size to be reliable and merging the blobs (in each of the separate directions) that were reasonably close to each other, we finished by using a Gaussian filter to smooth the segmentation result.

At the colorization stage, the intersection of the texture blobs with each of the directions is found and the region is colored based on the color value corresponding to the particular data variable being

represented over that region. Best results are achieved when the directions selected for in the segmentation correspond to perceptually dominant directions in the texture pattern. For example, one can see that pattern d052 consists of intrinsic parts that follow four different directions, while pattern d064 has two perceptually prominent directions (vertical and horizontal), as does pattern d01. When applicable, each of the oriented regions extracted from a pattern can be subdivided into smaller subsections to allow the representation of more colors. An example is of such a situation is shown in Figure 3.



**Figure 3.** Interweaving colors over four and six different regions in patterns d052 (top) and d064 (bottom), respectively, from the Brodatz album [1966].

#### 4 Applications of Texture Colorization

In this section we present some results in which we use combinations of color and texture to represent multiple variables on a 2D map. Figures 4–6 show a more dataset consisting of five different scalar distributions. We use the methods described in section 3 to produce a unified map in which the presence or absence of a significant quantity of each of the five different variables is represented across each region. One variable is mapped to the choice of texture pattern (d052 or d064) and the other four variables are mapped to the presence or absence of colors that follow the vertical, horizontal and/or diagonal directions.

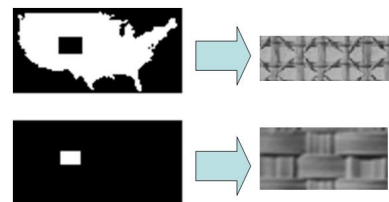
It is necessary to note that the choice of texture to be used within each of these areas was made based on the number of variables that we would eventually want to display in each of the regions that were to be filled with the texture in question. For example, for the data shown in figures 4–6, because a close inspection of the four variables to be represented by oriented components revealed that only two of these variables had values above the threshold in the smaller area, we determined that it was possible to use pattern d064 in that area, even though this pattern only contains two different perceptually salient directions. However, because we would need to simultaneously represent significant

quantities of four different variables at certain places in the larger area, we needed to use a pattern that has four different perceptual direction components for filling that area. In all areas of the final map, the color blue (representing trait 3) follows the vertical direction of the texture, trait 4 is shown with the color green following the horizontal direction, and the colors red and black, showing traits 5 and 6, follow the 45° and 135° directions respectively.

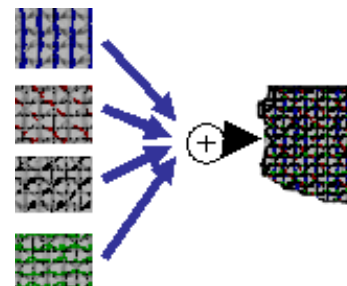
Clearly we could easily have used artificial line textures in place of d064 and d052. However, by providing an automatic technique for incorporating color into nearly any arbitrary texture, we give the visualization designer the freedom to choose from among a far greater variety of aesthetically pleasing multidirectional patterns, with the added benefit that the designer is free to use different texture patterns over different regions of the map.

#### 5 Discussion

The techniques described in this paper are suitable for visualizing any kind of data which naturally covers an area, such as data defined over a 2D map. By choosing to display the multiple colors representing the values of multiple co-located distributions side-by-side, rather than as a simple color mixture, this method inherently requires that a finite *area* of pixels be available for the representation of a set of data values at a point, rather than requiring a minimum footprint size of only one pixel. In order to successfully use texture for data visualization, it is important to choose a pattern whose spatial frequency is relatively high in comparison with the spatial frequency of the variations in the values of the data being represented. By using a finer texture we can more accurately convey detailed information in smaller regions. In addition, a higher resolution display may be helpful when using these techniques to visualize data in which many small regions are present. This is particularly the case in applications such as geophysics, in which scientists routinely rely on printed maps that are two to three feet tall to visualize their data.



**Figure 4:** Trait 1 and trait 2 are distinguished by texture type in this map. The area corresponding to Kansas is subsequently colored in two directions, to show traits 3 and 4. The larger area is colored in four directions, to show traits 3, 4, 5 and 6.



**Figure 5.** An area of the final map in which 5 variables overlap (four colors and the texture d052).



## 6 Conclusions and Future Work

In this paper we have discussed several different techniques for easily and automatically compositing color into a texture pattern for the purposes of multivariate visualization. In particular we showed how, through conventional and widely used algorithms, it is possible to map different colors to different oriented components of an existing texture image, in order to visualize multiple overlapping data distributions. We provided several examples based on real data obtained from cartographic maps. By compositing color with texture, we can increase the number of co-located variables that can be effectively visualized, beyond what would be possible using either just texture or just color compositing alone. Although using a combination of texture and color sometimes is not the most effective way to visualize all types of data, we do believe that these methods can be particularly useful when one needs to visualize a handful of different variables over a map-like domain containing relatively large uniformly valued regions. In future work, we would like to find a quantitative method for choosing the optimum level of color saturation and value that can be used along with textures without affecting the accuracy of perception among different variables.

### References

- Brewer, Cynthia A. Color Use Guidelines for Mapping and Visualization, In *Visualization in Modern Cartography*, edited by A.M. MacEachren and D.R.F. Taylor, Elsevier Science, Tarrytown, NY, 1994, pp. 123–147.
- Brodatz, Phil. *Textures: A Photographic Album For Artists and Designers*, Dover, 1966.
- Gossett, Nathan and Chen, Baoquan. Paint Inspired Color Mixing and Compositing for Visualization. In *IEEE Symposium on Information Visualization* (2004).
- Healey, Christopher G. Choosing Effective Colours for Data Visualization, *Proc. IEEE Visualization '96*, pp. 263–270.
- Healy, Christopher and Enns, James. Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization. In *IEEE Transactions on Visualization and Computer Graphics* (1999), vol. 5, pp. 117–125.
- Kiu, M. and Banks, D. C. Multi-Frequency Noise for LIC. In *Proceedings of IEEE Visualization '96*, pp. 121–126.
- Levkowitz, Haim. Color Icons: Merging Color and Texture Perception for Integrated Visualization of Multiple Parameters, In *Proc. IEEE Visualization '91*, pp. 164–170.

- Liu F. and Picard R. Periodicity, Directionality and Randomness: Wold Features for Image Modeling and Retrieval. In *IEEE Transactions on Pattern Analysis And Machine Intelligence* (1996).
- Rao, A.R. and Lohse G.L. Identifying High Level Features of Texture Perception. In *Graphics Models and Image Processing* (1993), vol. 53, pp. 218–233.
- Rao, A.R. and Lohse G.L. Towards a Texture Naming System: Identifying Relevant Dimensions of Texture. In *IEEE Visualization '93*, pp. 220–227.
- Robertson, Philip K and O'Callaghan, John F. The Generation of Color Sequences for Univariate and Bivariate Mapping. In *IEEE Computer Graphics and Applications*, 6(2), February 1986 pp. 24–32.
- Rogowitz, Bernice E. and Treinish, Lloyd. Data Visualization: The End of the Rainbow, In *IEEE Spectrum*, pp. 52–59, December, 1998
- Sanderson, A., Johnson, C., Kirby, R. Display of Vector Fields Using a Reaction-Diffusion Model. In *Proceedings of IEEE Visualization 2004*. pp 115–122.
- Simoncelli, E.P. and Freeman W.T. The Steerable Pyramid: a Flexible Architecture for Multi-Scale Derivative Computation. In *IEEE Second International Conference On Image Processing* (1995).
- Slocum, T. *Thematic Cartography and Visualization*, Prentice-Hall, 1999.
- Tamura, H., Mori, S. and Yamawaki, T. Texture Features Corresponding To Visual Perception, In *IEEE Transactions On Systems, Man, and Cybernetics*, 8(6): 460–473, 1978.
- Urness, T., Interrante, V., Longmire, E., Marusic, I., and Ganapathisubramani, B.: Effectively Visualizing Multivalued Flow Data Using Color and Texture. In *Proceedings of IEEE Visualization* (2003), pp. 151–121.
- Van Wijk, J. Spot Noise-Texture Synthesis for Data Visualization. In *Proceedings of ACM SIGGRAPH* (1991), pp. 309–318.
- Ware, C. and Knight, W. Orderable Dimensions of Visual Texture for Data Display: Orientation, Size, and Contrast. In *ACM Conference On Human Factors In Computing System* (1992), pp. 203–209.

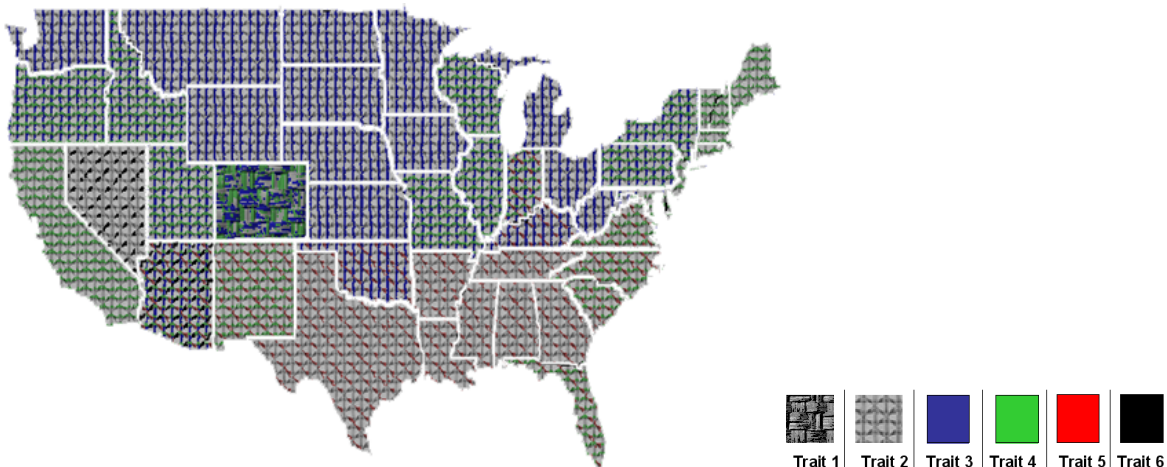


Figure 6: Visualizing multivariate data using the presented techniques.