1-1-2009

Hyperspectral Image Visualization Using Double And Multiple Layers

Shangshu Cai

HYPERSPECTRAL IMAGE VISUALIZATION USING

DOUBLE AND MULTIPLE LAYERS

By

Shangshu Cai

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Computer Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

May 2009

HYPERSPECTRAL IMAGE VISUALIZATION USING DOUBLE AND MULTIPLE LAYERS

By

Shangshu Cai

Approved:

Robert J. Moorhead II
Professor of Electrical and Computer Engineering
(Major Professor)

Jenny Q. Du
Associate Professor of Electrical and Computer Engineering
(Dissertation Director)

J. Edward Swan II.
Associate Professor of Computer Science and Engineering
(Committee Member)

T. J. Jankun-Kelly
Assistant Professor of Computer Science and Engineering
(Committee Member)

James E. Fowler
Professor of Electrical and Computer Engineering
(Committee Member and Interim Graduate Program Coordinator)

Sarah A. Rajala
Dean of the Bagley College of Engineering

Name: Shangshu Cai

Institution: Mississippi State University

Major Field: Computer Engineering

Major Professor: Dr. Robert J. Moorhead II

Dissertation Director: Dr. Jenny Q. Du

Title of Study: HYPERSPECTRAL IMAGE VISUALIZATION USING DOUBLE AND MULTIPLE LAYERS

Pages in Study: 117

This dissertation develops new approaches for hyperspectral image visualization. Double and multiple layers are proposed to effectively convey the abundant information contained in the original high-dimensional data for practical decision-making support. The contributions of this dissertation are as follows.

1. Development of new visualization algorithms for hyperspectral imagery. Double-layer technique can display mixed pixel composition and global material distribution simultaneously. The pie-chart layer, taking advantage of the properties of non-negativity and sum-to-one abundances from linear mixture analysis of hyperspectral pixels, can be fully integrated with the background layer. Such a synergy enhances the presentation at both macro and micro scales.

2. Design of an effective visual exploration tool. The developed visualization techniques are implemented in a visualization system, which can automatically preprocess and visualize hyperspectral imagery. The interactive tool with a user-friendly interface will enable viewers to display an image with any desired level of details.

3. Design of effective user studies to validate and improve visualization methods. The double-layer technique is evaluated by well designed user studies. The traditional approaches, including gray-scale side-by-side classification maps, color hard classification maps, and color soft classification maps, are compared with the proposed double-layer technique. The results of the user studies indicate that the double-layer algorithm provides the best performance in displaying mixed pixel composition in several aspects and that it has the competitive capability of displaying the global material distribution. Based on these results, a multi-layer algorithm is proposed to improve global information display.

ACKNOWLEDGMENTS

I acknowledge with sincere gratitude the support, guidance, and assistance provided by my major advisor Dr. Robert Moorhead and my dissertation director Dr. Jenny Q. Du. Without their invaluable help, this dissertation could not be materialized.

I thank my committee members, Dr. J. Edward Swan II, Dr. T.J. Jankun-Kelly, and Dr. James E. Fowler for their guidance and fruitful discussion.

And also, I would like to thank all the participants in user studies. Their participation is important to the success of this research.

I am grateful to my lab mates: Dr. Zhanping Liu, Dr. Phil Amburn, Derek Irby, Joel P. Martin, Mahnas Jean Mohammadi-Aragh, John van der Zwaag, Jibonananda Sanyal, Keqing Wu, Nareenart Raksuntorn, He Yang, and Wei Zhu. It is pleasure to work with them. I also want to thank my girlfriend, Hualu Zheng, and my roommate, Mingbo Luo, and many other friends. In the past years, they offered me great friendship and help.

Finally, special gratitude goes to my parents and my family for their support and encouragement.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Hyperspectral imaging, also known as imaging spectrometry, is an emerging technology in remote sensing. It uses hundreds of co-registered spectral channels to acquire images of the same area on the earth. The resulting hyperspectral imagery has very high spectral resolution, providing better diagnostic capability for object detection, classification, and discrimination than the traditional multispectral imagery. However, it is challenging to display the useful information contained in such a huge three-dimensional (3D) data cube. A common practice is to provide a presentation for a quick overview of a scene to support decision-making. Obviously, displaying a single image results in the loss of information. In this research, the objective is to maximize the information transfer in the display creation process.

Hyperspectral imagery can be visualized after dimension reduction, such as a three-band Red-Green-Blue (RGB) combination, principal component analysis (PCA) [6][9][10], and band fusion [41][42][43]. However, these methods result in significant information loss. Moreover, our study shows that any transform-based preprocessing without class separability enhancement will yield a poor display where different objects cannot be visually separated. Therefore, to preserve the maximum amount of information, it is preferred to visualize the classification result.

The high dimensionality of a hyperspectral image enables spectral unmixing at the subpixel level (subpixel level unmixing refers to the classification of co-existing materials at a pixel by a classification algorithm). Linear mixture analysis (LMA) [17][20][35] is a well-known technique in remote sensing. It can estimate the abundances of pure materials in each pixel. An unmixing result provides a soft classification map, describing the material distributions in an image scene. Hyperspectral image visualization can be achieved by effectively displaying these classification maps. It is noteworthy that abundances in each pixel satisfy the non-negativity and sum-to-one constraints. This enables the use of a pie-chart to show the detailed pixel composition. As for the overall material distribution, an optimal or suboptimal color combination can be applied to fuse all the classification maps. The resulting double-layer technique (denoted as DBLY) can display the material distribution both locally and globally with a controlling parameter adjusting the desired level of visible details.

The visualization of the classification result can be considered a multivariate visualization problem, where discrete and continuous visual elements are combined to visualize multiple fields on a surface. For instance, oriented slivers were employed to visualize multivariate data in [27][28]; dots with different sizes and distributions were used to represent different variables in [31]. All of these methods take the multiple variables as independent variables, and no relationship exists among them because they are different physical quantities. However, the concepts can be adopted after modification by considering the unique aspects of hyperspectral imagery. Thus, in addition to the aforementioned DBLY, a data-driven spot (DDS) layer can be introduced to further manifest the widely distributed materials in an image scene, and an oriented sliver (OS) layer can be employed to emphasize the distribution of important anomalous objects. The resulting multi-layer visualization technique can display the overall material distribution, widely distributed materials, anomalous objects, and subpixel level composition within a unified interface.

The DBLY visualization technique was evaluated via user studies. It was compared to traditional approaches, such as pseudo-color composites and gray-scale classification maps. In addition, the developed and refined techniques were implemented in a visualization system, which has the capability of automatically preprocessing and visualizing hyperspectral imagery. The interactive tools in a user-friendly interface enable the display with the details at the discretion of a viewer.

The contribution of this research is the development of versatile methods to present the information contained in a hyperspectral image. It takes advantage of the LMA in the remote sensing area and the multivariate visualization techniques in the visualization area. Obviously, this synergy produces a new multidisciplinary research area full of promise.

The dissertation is organized as follows. Chapter II reviews the current existing visualization techniques and their limitations. Hyperspectral data preprocessing steps are discussed in Chapter III. The DBLY technique is presented in Chapter IV. Chapter V and Chapter VI discuss the user studies on synthetic and real datasets. Chapter VII discusses the user studies and DBLY. Chapter VIII draws conclusions and presents future work.

CHAPTER II

BACKGROUND

This chapter briefly describes the features of hyperspectral data, the visualization techniques that have been used to visualize remotely sensed imagery, and their limitations. Other potential visualization techniques that can be used to visualize hyperspectral imagery are also introduced.

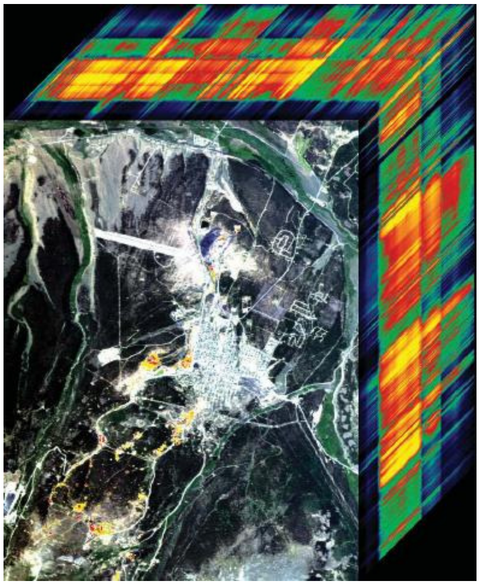


Figure 2.1 A hyperspectral image cube with 220 spectral bands [12].

2.1 Hyperspectral Imagery

A hyperspectral sensor acquires images with hundreds of spectral bands. Its high spectral resolution provides the capability of more accurate material detection and classification. A hyperspectral image can be viewed as a 3D data cube. Figure 2.1 shows Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) hyperspectral imagery of the Leadville mining area in Colorado [12].

2.2 Remotely Sensed Imagery Visualization

Visualization techniques were first introduced to display remotely sensed imagery soon after the first remote sensor was used to acquire data. The traditional gray-scale visualization technique is still used to display remotely sensed images. Since a gray-scale image can display only one band, true color composite images are created by using red, green, and blue-green bands for an RGB color display [3]. The color infrared (CIR) composite, an important approach to display the distribution of vegetation [1], maps the green, red, and near-infrared bands into the three RGB channels. Obviously, selecting only three bands to form a color image does not display all the information contained in a hyperspectal image. In [4], Robertson *et al*. employed a transform to project the original bands into three channels and then mapped the three channels to a perceptual color space to achieve a high contrast image. Durand *et al.* enhanced a color contrast display by balancing the Signal-to-Noise-Ratio (SNR) of the three bands [5].

For hyperspectral imagery, PCA is a widely used technique to reduce its dimension. It projects the original data into orthogonal principle components (PCs). Most of the energy is in the first several PCs; therefore it is assumed that the first several PCs include the most information in the original dataset and they can be used to display the hyperspectral images. Tyo *et al.* visualized the hyperspectral imagery by mapping the first three PCs to the HSV color space [6]; segmented PCA was employed by Vassilis *et al.* to fuse the hyperspectral images [7]. However, the rank of a PC may be affected by the noise level; thus a high-ranked PC may contain less information than a low-ranked PC if the noise contribution is significantly large. Moreover, only choosing the first several PCs may cause information loss because some objects may be present in lower ranked PCs than the first three PCs. Therefore, a noise-adjusted principal component analysis (NAPCA) and interference-and-noise-adjusted principal component analysis (INAPCA) [8] may be better choices. A PCA-class method for color display has been presented for hyperspectral image display in [9][10]. Instead of using a transform to compact the information, Jacobson *et al.* combined all the bands in a hyperspectral image using fixed linear spectral weights for each channel and visualized the result in a natural-looking image [11]. This method can consistently display the same material in different datasets in a similar color.

Hyperspectral image analysis is task-driven [12]. For most tasks, e.g. detection or classification, the materials need to be separated as clearly as possible. A visualized image cannot have high quality if a transform does not enhance the class separability. Therefore, visualizing the detection or classification result generally can maximize the information to be conveyed. In the hard classification, where a pixel is assigned to only one class, a color display is created by assigning different colors to different classes. An automatic color selection system was implemented by Campadelli *et al.* [13]. Marcal presented an automatic color index algorithm to display the results of hierarchical classification [14]. Due to the low spatial resolution, pixels in multispectral or hyperspectral images are mixed [15]. This problem can be addressed by a linear mixture model (LMM) [16][17], which assumes a pixel vector is the linear combination of pure materials, called endmembers, in an image scene. LMM is used to classify mixed pixels [18][20][21]. In mixed pixel classification, a pixel is classified as a certain percentage (i.e., abundance) of each material. How to display the mixed classification results is a challenging problem. One of the traditional ways is to view each endmember abundance map (i.e., classification map) as a gray-scale image side-by-side. There are at least two drawbacks for viewing each endmember as a separate gray-scale image: first, it is difficult for viewers to understand the overall distribution of all the materials in the image scene [22]; second, it is difficult to show the spatial relationship among classes. Wessels *et al.* tried to conquer this problem by displaying abundance maps in one image, where a pixel was displayed in the color of the endmember with the largest abundance in this pixel [24]. This causes severe distortion because materials with smaller abundances were suppressed.

2.3 Multivariate Visualization

Visualizing classification results can be taken as a multivariate visualization problem. This problem has been studied for many years and it remains as one of the hottest research topics in visualization [2].

Texture is widely used to display information. Bair *et al.* studied the Gabor model and suggested the basic dimensions of textures, such as orientation, size, contrast, etc., can be used to present information [38]. Crawfis used textured splats to visualize multiple scalar and vector fields [25]. Forsell *et al.* employed texture and 3D surface shape to display multivariate data simultaneously [26]. In Urness's work multiple flow fields and scale variables were visualized by texture, glyph, embossing, and color [29]. Synthesized Cell Texture (SCT) was presented by Vickery to display multiple fields [32]. Interrante employed natural textures and successfully mapped multivariate data on the same map [36]. Oriented sliver texture was presented by Weigle *et al.* [27] and Healy [28][54] to visualize multiple collocated fields.

Glyphs are another information display technique. In [30], Kirby *et al.* mapped multiple flow variables to different visual elements and multiple fields were effectively visualized on a surface. Bokinsky showed that different sizes and distributions of dots could successfully represent different variables [31]. Boxes and arrows were placed at regular intervals to display the uncertainties by Schmidt *et al.* [37]. Levkowitz harnessed color icons for visualization [40].

These existing techniques successfully visualize the collocated multiple vector or scale fields in one single image, but they only visualize the qualitative information and they still have the problem of how to represent quantitative information precisely. In addition, they take the multivariate data as independent variables. However, this assumption is not valid for hyperspectral image because the spectrums of endmembers are measuring the same area. Therefore, new approaches need to be developed to take the peculiarities of hyperspectral classification into consideration.

2.4 Uncertainty and Accuracy in Visualization

Visualization is often utilized by researchers to explore datasets in order to support practical decisions. In general, visualization processing is based on two assumptions: the visualized data is accurate and the visualization processing is free from errors. However, these two assumptions may not be true for all cases. Uncertainties widely exist in data collection and in visualization processing. Many researchers have discussed how to visualize the uncertainties in data and its processing procedures in recent years.

The uncertainties that exist in remote sensing data may cause users to make a wrong decision. Bastin *et al.* stated that an ideal visualization map should be accompanied with uncertain descriptions; fuzzy classification was employed to handle the potential uncertainties in satellite data [62]. An interactive map was created by Dungan *et al.* to visualize the uncertainties in remote sensing [63]. Digital Elevation Model Error Viewer (DEMEV) was introduced by Gouse *et al.* to assist the analysis and visualization of possible errors in Digital Elevation Models (DEMs) [64].

Large-scale uncertainties in astrophysical environments were effectively visualized by Li *et al.* [65]. Lodha *et al.* presented UFLOW—a system to visualize the uncertainties or errors caused during data acquirement, transformation, and rendering in fluid flow [66]. Probabilistic animation was introduced by Lundstrom *et al.* to visualize the uncertainties in medical volume data [67].

Inaccuracy is not caused during data collection only; it can be also introduced by visualization processing. In order to analyze the errors which are caused by interpolation in visualization processing, both interpolation function errors and field data errors inside a tetrahedral cell were visualized by Doi *et al*. [68]. Grigoryan *et al*. presented an algorithm to visualize surfaces with uncertainties by using point-based rendering [69]. An interactive scheme was provided by Harzo *et al*. to compare the difference of cosmology simulations [70]. Lopes presented several algorithms to improve the accuracy of contouring, isosurfacing, and particle tracing [75]. In order to understand the errors in visualization, researchers began to utilize user studies to quantitatively measure the errors caused by visualization methods.

2.5 Evaluation and User Study

There is a long history of user studies being utilized to validate techniques. For instance, Bly and his colleagues employed a user study to compare tiled and overlapping windows in the 1980s [71]. Because a user study can evaluate the effectiveness and weaknesses of techniques, they have been widely used to quantitatively measure visualization algorithms in recent years. Laidlaw *et al*. compared six techniques for visualizing 2D flow fields and measured user performance on three flow-related tasks for each of the six algorithms [55]. Critical point classification and recognition, and symmetric pattern classification tasks were designed by Liu *et al*. to compare the existing flow visualization techniques [76]. Hagh-Shenas *et al*. compared two alternative algorithms for visualizing multiple discrete scalar datasets with color [58]. Azuma *et al*. evaluated the different label placement strategies for augmented reality view management by traditional statistical analysis and an empirical user study [61]. An empirical study was used to compare the performance of two techniques which visualized task relationships and temporal uncertainties in software engineering project planning [72]. Pausch *et al*. conducted a user study to compare head-mounted and stationary displays [73].

A user study can not only validate visualization algorithms, but also provide guidelines of new visualization techniques. A well-conducted user study can significantly improve the research quality [57]. With a user study, Healey built several basic rules for choosing color effectively to visualize multivariate data [54]. Ward and Theroux conducted a user study to find the optimal viewing for layered texture surfaces [56]. Acevedo *et al*. investigated how the perceptual interactions among visual elements, such as brightness, icon size, etc., affect the efficiency of data exploration based on a set of 2D icon-based visualization methods [39]. Martin and his colleagues found that participants improved their performance as more objects were rendered in the scene when they conducted a hurricane related user study [74]. Bair *et al*. identified three phases of a user study: defining goals, creating datasets, and performing studies [38]. In this dissertation, we present two well designed user studies based on synthetic and real datasets to verify the efficacy of the proposed algorithms for hyperspectral image visualization.

CHAPTER III

DATA PREPROCESSING

Data preparation is necessary for the visualization process. A hyperspectral image may contain some bad bands, such as water absorption and low SNR bands, and bad pixels, such as white spots due to sensor saturation. In order to reduce the impact from bad bands and bad pixels, they need to be removed before classification. This chapter discusses the details of automated bad band and bad pixel removal. The linear mixture model-based classification will also be introduced.

3.1 Bad Band Removal

Two adjacent bands in a hyperspectral image tend to have very high correlation. The bands that are not similar to their neighbors may contain a high level of noise. Water molecules absorb portions of the spectrum while electromagnetic radiation transmits through a medium containing water molecules; therefore, water absorption bands have low reflectance and noise plays a key role in these bands. The correlation between adjacent water absorption bands is also very low. These facts are used to automatically remove all these unwanted bands. The detailed steps are as follows:

1. Compute the correlation coefficient between each pair of adjacent bands, which is denoted as $\rho_{l(l+1)}$ for $0 < l < L$, where L is the number of bands.

2. For the l-th band, compare the correlation coefficients $\rho_{(l-1)l}$ and $\rho_{l(l+1)}$ with a given threshold η. If both of them are greater than η, this band is kept; otherwise, it is removed as a bad band.

Table 3.1 displays the bad band removal result for three data sets with the threshold $\eta = 0.8$. The results are very close to the manually selected bands. The high quality of automatic band selection demonstrates that this approach is an effective bad band removal algorithm.

Table 3.1

NUMBER OF BANDS BEFORE AND AFTER SELECTION

Dataset	Original	After Selection
AVIRIS Lunar Lake	224	158
AVIRIS Low Altitude	224	132
Hyperion	220	152

3.2 Bad Pixel Removal

Due to sensor noise, some bad pixels may occur in a selected band. Two types of noise may exist: isolated pixels and stripes. Normally, the isolated pixels are caused by over saturation; and stripes are caused by dead detectors in push-broom sensors. An isolated noisy pixel is not spatially or spectrally correlated with neighboring pixels. The pixels in a noisy line are correlated. Thus different algorithms are used to remove isolated pixels and lines.

3.2.1 Isolated Noisy Pixel Removal

A low pass filter is often used to smooth out isolated noisy pixels, but it will blur the normal pixels. An isolated noisy pixel removal method should keep the normal pixels when it is fixing or removing a bad pixel.

Bad pixels are pixels whose value is drastically dissimilar to their neighbors. Bad pixels may be caused by a bad sensor detector or by sensor saturation. This implies that a bad pixel may be removed by a local relationship (i.e., spatial and spectral neighboring pixels). An algorithm with the following steps can be applied.

1. Find the maximum pixel value in a band.
2. Find the pixels whose values are greater than a percentage (λ_1) of the maximum value and add these pixels to the abnormal pixel candidate set.
3. Check the value of abnormal pixel candidates with their adjacent bands. If the pixel value exceeds the maximum value of the same pixel (spatial position) in the adjacent bands by a percentage (λ_2), it is considered as a bad pixel.
4. Conduct the signature interpolation and use the interpolated values to replace the bad pixels.

In our experiments, $\lambda_1 = 80\%$ and $\lambda_2 = 90\%$ achieved good results. The spectral inspection in Step 3 is particularly important to avoid accidentally removing those normal pixels with large reflectance. Figure 3.1 displays Band 173 in the AVIRIS Low Altitude imagery before and after bad pixel removal. Figure 3.1(a) is dark because some isolated pixel values are so large that they suppress the brightness of other pixels. Figure 3.1(b) is the image after the bad pixels have been removed, where image content becomes visible.

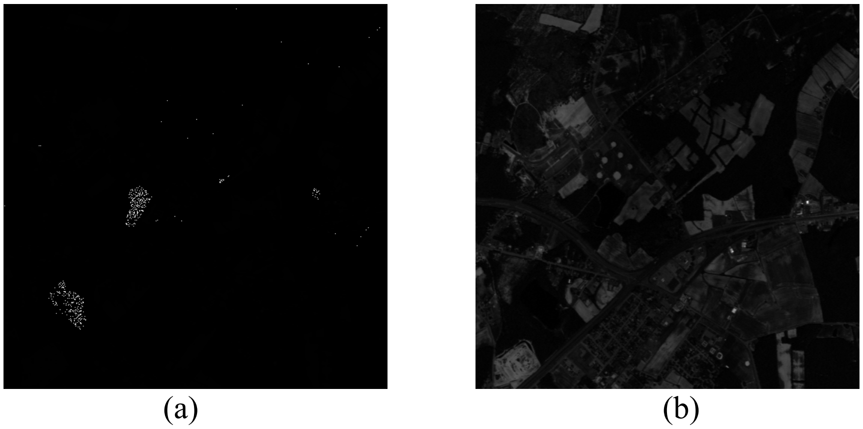


Figure 3.1 Band 173 in the AVIRIS Low Altitude data. (a) Before bad pixels are removed; (b) after bad pixels are removed.

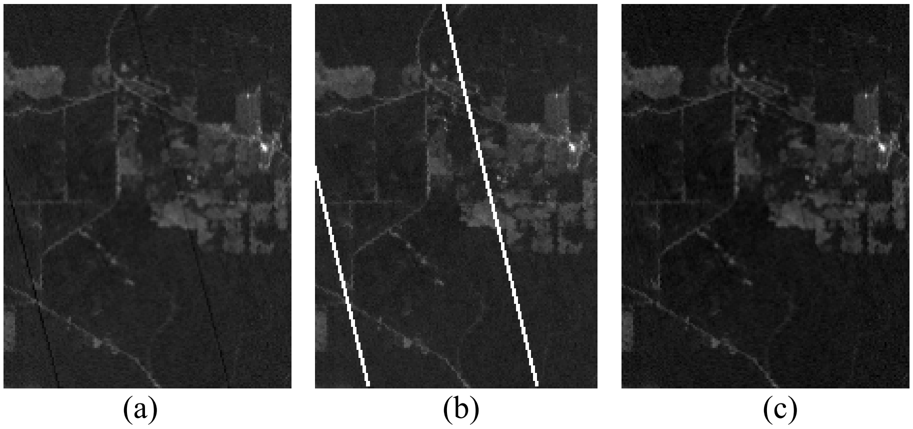


Figure 3.2 Band 10 of Hyperion data with the line removal. (a) Original image; (b) detected stripes; and (c) after the stripes removed.

3.2.2 Dark or White Noisy Line Removal

A stripe pattern is one of the common noise types in push-broom sensed images [45], and it is caused by bad sensor detectors. The algorithm is developed to automatically remove a dark line pattern with the Hough transform [46]. The algorithm is described as follows:

1. Compare each pixel with its four closest neighbors. If its value is less than ε, say, 80%, of the neighborhood average, mark this pixel as a bad pixel candidate.
2. Create a binary image where the pixels corresponding to the bad pixel candidates are set to 1 and others 0.
3. Detect lines across the entire binary image using the Hough transform. The candidates that are located in the detected lines are considered as bad pixels.
4. Interpolate and replace the bad pixels along the detected lines.

This algorithm has been tested on a Hyperion dataset. The results show that all visible stripes can be automatically removed after applying this algorithm. Figure 3.2 shows Band 10 with two dark lines. The line patterns can be effectively detected and removed. Because a white stripe pattern is similar to a dark pattern, this algorithm can be easily modified to detect white line noise by changing several parameters.

3.3 Linear Mixture Model

Detailed information about a pixel in a hyperspectral image is of great interest for both civilian and military purposes. The reflectance of pixels can be considered as the result of a linear combination of pure materials, called endmembers. So pixels are mixed. The linear mixture model (LMM) has been widely used to analyze hyperspectral imagery [18][20], [33][34].

Let a pixel vector in a hyperspectral image be represented as $\mathbf{r} = [r_1, \cdots, r_l, \cdots, r_L]$, where L is the number of the bands. Assume p is the number of endmembers in the scene and their signatures are denoted as $\mathbf{M} = [\mathbf{m}_1, \cdots, \mathbf{m}_k, \cdots, \mathbf{m}_p]$, where $\mathbf{m}_k$ is an $L \times 1$ vector corresponding to the signature of the *k*-th endmember. The $\mathbf{r}$ can be represented as:

$$\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n} \tag{3.1}$$

where $\boldsymbol{\alpha} = \left(\alpha_1 \cdots \alpha_k \cdots \alpha_p\right)^T$ is an $p \times 1$ column abundance vector, whose *k*-th element represents the abundance of the *k*-th endmember material in the pixel $\mathbf{r}$. Here, $\mathbf{n}$ represents system noise or sensor measurement error.

Since $\boldsymbol{\alpha}$ represents abundances, α_k for $1 \le k \le p$ should satisfy two constraints [20]: all abundances should be non-negative (referred to as non-negativity constraint), and the sum of all abundances in a pixel should be one (referred to as sum-to-one constraint). These two constraints can be formed as:

$$\sum_{k=1}^{p} \alpha_k = 1 \text{ and } 0 \le \alpha_k \le 1 \tag{3.2}$$

The LMA can be divided into two cases: $\mathbf{M}$ is known and $\mathbf{M}$ is unknown. For the first case, we have the prior knowledge about the endmember signatures and want to estimate the $\boldsymbol{\alpha}$, which can be solved by a least squares approach. An unconstrained least squares problem can be formulated to minimize the reconstruction error: $(\mathbf{r} - \mathbf{M}\hat{\boldsymbol{\alpha}})^2$, where $\hat{\boldsymbol{\alpha}}$ is calculated by:

$$\hat{\boldsymbol{\alpha}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{r} \tag{3.3}$$

The fully constrained least squares linear unmixing (FCLSLU) with the two constraints in Eq. (3.2) being relaxed can be solved by quadratic programming. The details of quadratic programming can be found in [19].

For the second case, both $\mathbf{M}$ and $\boldsymbol{\alpha}$ need to be estimated, where the unsupervised fully constrained least squares linear unmixing (UFCLSLU) can be applied [20]. The algorithm can be described as:

1. Select the two pixels with the maximum and minimum norm from the hyperspectral imagery and construct $\hat{\mathbf{M}} = [\mathbf{m}_1, \mathbf{m}_2]$. Then use the quadratic programming to solve $\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \hat{\alpha}_2)^T$.
2. Calculate the reconstruction error, e, between the pixel vector $\mathbf{r}$ and its estimate, i.e., $e = \left\| \mathbf{r} - \hat{\mathbf{M}}\hat{\boldsymbol{\alpha}} \right\|$.
3. Find the pixel that has the maximum error and take it as the third endmember, i.e., $\hat{\mathbf{M}} = [\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3]$. This pixel is selected because it is considered the most dissimilar pixel from $\mathbf{m}_1$ and $\mathbf{m}_2$.
4. Repeat Steps 2 and 3 for additional endmembers and their abundances, until the error is less than a given threshold ξ or the maximum number of endmembers is reached.

When the number of endmembers is unknown, a large number can be assumed initially to run the UFCLSLU algorithm. Then the similar endmember signatures can be combined after the similarity comparison using the spectral angle mapper (SAM) technique [43], and the endmember signatures corresponding to noisy abundance images with large entropies can be removed. The remaining signatures are used for the supervised FCLSLU to generate the final abundance images for visualization.

The AVIRIS Lunar Lake data shown in Figure 3.3 is classified by the UFCLSLU algorithm. After bad band removal, 158 out of 224 bands were left for LMA. Figure 3.4 shows the six material abundances, named {Playa Lake, Rhyolite, Vegetation, Anomaly, Cinder, Shade} based on some prior knowledge [44]. In the gray-scale abundance images, a white pixel represents a high abundance of the corresponding endmember.

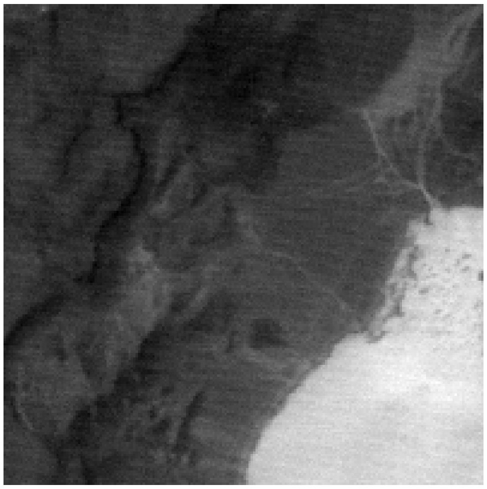
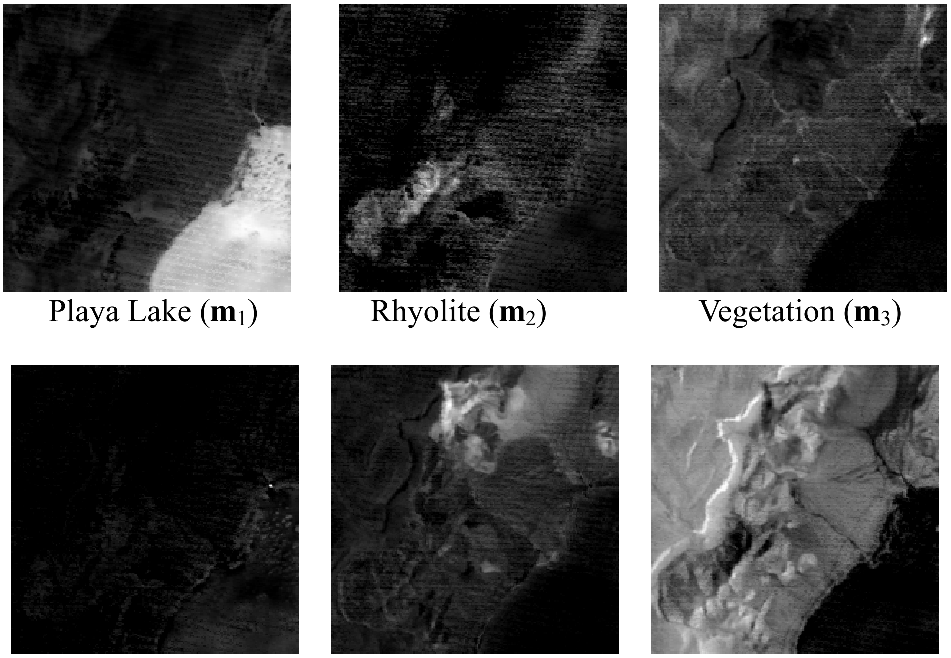


Figure 3.3 AVIRIS Lunar Lake scene of size 200 × 200.



Playa Lake ($\mathbf{m}_1$)

Rhyolite ($\mathbf{m}_2$)

Vegetation ($\mathbf{m}_3$)

Anomaly ($\mathbf{m}_4$)

Cinder ($\mathbf{m}_5$)

Shade ($\mathbf{m}_6$)

Figure 3.4 The abundance images of the AVIRIS Lunar Lake scene.

CHAPTER IV

DBLY VISUALIZATION

Traditionally, the resulting abundance images from LMA are displayed as gray-scale images. There are at least two drawbacks: 1) it is difficult for viewers to find the overall distribution of the *p* materials in the image scene; 2) it is difficult for viewers to know the material composition at any specific location in the image scene.

In this chapter, a new approach is introduced to visualize the *p* abundance images in a single color image by employing DBLY. Layer I displays the general distribution of materials and Layer II displays the detailed composition of each pixel. The final image display is generated by overlaying Layer II on Layer I.

4.1 Color Assignment Strategies

Because color is intuitive, color is widely used in visualization to indicate different materials or values. Appropriate color selection helps viewers discriminate different materials easily and quickly. In this research, an automatic color selection approach was developed to label different endmembers.

In the study of color representation, one of the first mathematically defined color spaces is the CIE XYZ color space, which was created by the International Commission on Illumination (CIE) in 1931 [79]. In CIE XYZ color space, colors can be represented by a linear combination of three basic spectral signatures (roughly red, green, and blue), which are taken as the three coordinates X, Y, and Z. The CIE XYZ color system simplifies the color representation; but it does not give a direct way of estimating color differences. The color distance in the CIE XYZ color space does not correspond to the perceptual distance. It would be desirable if a distance on a chromaticity diagram is well correlated to the degree of difference between two colors. Two uniform color spaces provided by CIE are: CIE LAB and CIE LUV, where the perceptual distance approximately equals the Euclidean distance [47][48].

However, the difference and the appearance of color not only depend on the tristimulus, but also heavily depend on the adapted state of viewers, the constancy and contrast, and the temporal and spatial structure of the tristimulus. A real ideal uniform color space should involve these factors. Unfortunately, such an ideal color visual model is unavailable because these factors are very complex and the viewer's environments are not predictable [48].

Healey addressed three criteria for color selection: color category, color distance, and linear separation [28]. If the color labels suits the three criteria, the colors can be easily separated. We chose the Hue-Saturation-Value (HSV) space because the color category suggested are very similar to the hue used in the HSV color space. Small color labels selected from the HSV space can belong to different color categories. Though the HSV color space is not uniform, the colors in the slice are very close to being uniform. The color selection from the HSV space does not theoretically satisfy the three criteria but it is very close. However, in some cases, even when distinctive colors have been assigned, the final color display result may not be as good as expected because color perception is nonlinear. It is also suggested by other researchers that the color selection will have good performance if a luminance (or lightness) dimension is separated from the chromatic dimensions [49].

Therefore, we will conduct distinct color selection in the HSV color space with the goal that a color display should have the capability of conveying more information about the spatial relationship among objects or materials and make it easier to separate pixels into different classes [50]. In hyperspectral research, experienced analysts may have prior knowledge about the endmembers and want to assign a specific color to an endmember. For instance, one may want to use green to display healthy vegetation. In order to suit this case, we developed two automatic color assignment strategies for the visualization system: one with class similarity constraint only and one with class similarity and color consistency constraints.

4.1.1 Automatic Color Assignment with Class Similarity Constraint

The endmembers can be arranged in a sequence based on the signature similarity using a metric such as SAM. Without loss of generality, assume the endmembers are arranged according to signature similarity, i.e., $\mathbf{m}_k$ is more similar to $\mathbf{m}_{k+1}$ than to $\mathbf{m}_{k+2}$. To relax the similarity constraint in color assignment, the color for the *i*th abundance image is less similar to the (k+1)-th image than to the (k+2)-th.

In the HSV color space, hue is an angle between 0° and 360°, and each angle corresponds to a specific color tone. In order to choose the colors as distinctively as possible, the angle of hues are uniformly selected from [0°, 360°] for the p classes with the saturation and value being 1.0. The p color labels can be represented as:

$$\mathbf{CL}_q = \begin{cases} H_q = 360^\circ q/p \\ S_q = 1.0 \qquad\qquad q = 0,\cdots,p-1 \\ V_q = 1.0 \end{cases} \tag{4.1}$$

In order to separate similar classes as far as possible, the color labels are assigned in an interleaved way. Let $\mathbf{h}_k$ be the color vector in the HSV space for the k-th class, for $k = 1,\cdots,p$, and let the color labels assigned to $\mathbf{h}_k$ be denoted as $\mathbf{CL}q_k$. The index q_k can be determined by

$$q_k = \left(q_{k-1} + \lceil p/2 \rceil\right) \bmod p \tag{4.2}$$

if $\mathbf{CL}q_k$ has not been assigned. Otherwise,

$$q_k = \left(\left(q_{k-1} + \lceil p/2 \rceil\right) \bmod p\right) + 1. \tag{4.3}$$

Here $\lceil p/2 \rceil$ chooses the largest integer that is less than or equal to $p/2$. q_1 can be any number between (0, p–1), but in general $q_1 = 0$. Figure 4.1(a) shows the six color labels assigned to six classes using Eqs. (4.1-4.3) where the correspondences are $\mathbf{h}_1 \leftarrow \mathbf{CL}_0$, $\mathbf{h}_2 \leftarrow \mathbf{CL}_3$, $\mathbf{h}_3 \leftarrow \mathbf{CL}_1$, $\mathbf{h}_4 \leftarrow \mathbf{CL}_4$, $\mathbf{h}_5 \leftarrow \mathbf{CL}_2$, $\mathbf{h}_6 \leftarrow \mathbf{CL}_5$.

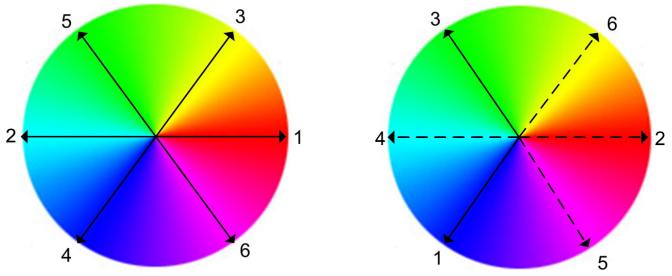
4.1.2 Automatic Color Assignment with Class Similarity and Color Consistency Constraints

In this color assignment method, the viewer can define the colors of some abundance images for consistent rendering. For instance, green is an obvious choice for healthy vegetation. If a standard signature library is available, the identities of some endmembers may be recognized and then their abundance images may be assigned to consistent colors if desired. The colors for the rest of the abundance images can be assigned with the similarity constraint. It should be noted that it may be impossible to satisfy both class similarity and color consistency constraints for the entire set of abundance images because these two constraints are contradictory in some cases.

Assume q $(0 < q < p)$ color labels have been predefined for q classes. The q color labels divide the 360° hue circle into q intervals $(\pi_1 \cdots \pi_w \cdots \pi_q)$, where π_w is the interval between $[H_w, H_{w+1}]$. Those classes whose colors have not been pre-defined are determined by the following rules:

1. Arrange the rest of classes in the order of similarity.
2. Pick the next class to assign the color vector **h**.
3. Find the maximum interval π_{wmax} and assign $H = (H_{wmax} + H_{wmax+1})/2$ as the hue component of **h**. If more than one interval has the maximum π, then choose the one that results in the H which is farthest from that for the previous class in the sequence.
4. Redivide the 360° hue slice into q + 1 intervals.
5. Repeat 2 to 4 until all the classes are assigned colors.

Figure 4.1(b) shows the color labels assigned to six classes where Class 1 and 3 are pre-defined as blue and green, respectively. After the colors have been assigned in the HSV color space, they are converted to the RGB color space so that they can be used in the next visualization steps and displayed on standard color monitors.



(a) (b)

Figure 4.1 Color assignment results. (a) Six classes are assigned color according to their signature similarity; (b) Class 1 and Class 3 are pre-defined as blue and green, respectively.

4.2 Layer I – Background Layer

Layer I, referred to as the background layer, is formed by pre-assigning a color, which is discussed in Section 4.1, to each of the *p* abundance images and merging them using a linear transformation. This layer gives the viewers the overall spatial distribution of endmembers.

4.2.1 Pixel Color Generation

Each pixel in the abundance images can be presented as a vector $\boldsymbol{\alpha} = \left(\alpha_1 \cdots \alpha_k \cdots \alpha_p\right)^T$ constrained by Eq. (3.2). The color that is selected by the color assignment strategies for the *i*th endmember can be denoted as $\mathbf{c}_k = \left(r_k\ g_k\ b_k\right)^T$ in the RGB color space. Then a color matrix can be constructed as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_k^T \\ \vdots \\ \mathbf{c}_p^T \end{bmatrix} = \begin{bmatrix} r_1 & g_1 & b_1 \\ \vdots & \vdots & \vdots \\ r_k & g_k & b_k \\ \vdots & \vdots & \vdots \\ r_p & g_p & b_p \end{bmatrix}. \qquad (4.4)$$

The final color for the vector $\boldsymbol{\alpha}$ in Layer I can be achieved by multiplying the pixel vector with the color matrix, i.e.,

$$\mathbf{c}^T = \boldsymbol{\alpha}^T \mathbf{C} . \tag{4.5}$$

Since $\boldsymbol{\alpha}$ was constrained by the non-negativity and sum-to-one constraints, the final color will be within the normal range [0, 1] of color components.

4.2.2 *Mapping Pixels to Vertices*

OpenGL is a device-independent 3D graphics library [51]. Setting a series of primitives in OpenGL can easily render 2D or 3D images. One of these primitives displays an OpenGL point with a size and color, which is referred to as a vertex. A vertex can be considered a geometrical point with a given spatial position and color. A simple mapping function is formed from an image pixel to an OpenGL vertex as

$$f : p_{ij} \rightarrow v_{ij} \qquad .(4.6)$$

Figure 4.2 illustrates the mapping procedure. The vertical and horizontal distance between two adjacent vertices is u. Then the mapping functions between p_{ij} and v_{ij} are represented as:

$$\begin{aligned} p_{ij}.\,position &= (i, j) \\ v_{ij}.\,position &= (iu, ju) \quad . \\ v_{ij}.\,color &= p_{ij}.\,color \end{aligned} \tag{4.7}$$

Based on this mapping function, the color of vertices is determined by the color of the corresponding pixel, which is obtained by Eq. (4.5). The color of any point that is not a vertex will be determined by linearly interpolating the colors of the four closest vertices. The color of any point in this layer, including vertices and the intervening space, is denoted as C_I.

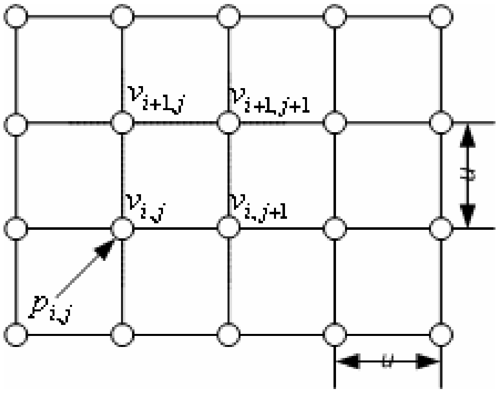


Figure 4.2 The mapping scheme from pixels to vertices.

4.3 Layer II – Detail Layer

Layer II, referred to as the detail layer, introduces a "superpixel" to display the distribution detail of each endmember in a pixel at that position. Superpixel, also known as circle segments [52] and star-maps [53], is a method to visualize multi-variant data.

In this layer, each endmember is represented by a fan-shaped region of a pie-chart, as shown in Figure 4.3. Without prior knowledge about the classified endmembers, each member is considered as equally important. Therefore, an endmember is randomly assigned to the *i*-th fan region. In general, the first endmember is chosen to assign the first region, and so on. The area of a fan-shaped region for the *k*-th endmember is proportional to the angle θ_k, which is determined by its abundance α_k, i.e.,

$$\theta_k = \alpha_k \cdot 360^\circ . \tag{4.8}$$

Its starting and ending positions can be represented as

$$\beta_k^s = \sum_{d=1}^{k-1} \theta_d \text{ and } \beta_k^e = \sum_{d=1}^{k} \theta_d \tag{4.9}$$

respectively. They can be related by $\theta_k = \beta_k^e - \beta_k^s$ and $\beta_1^s = 0°$. Because $\boldsymbol{\alpha}$ is constrained by non-negativity and sum-to-one constraints, a superpixel will be shown as a full disk, i.e., $\beta_p^e = 360°$.

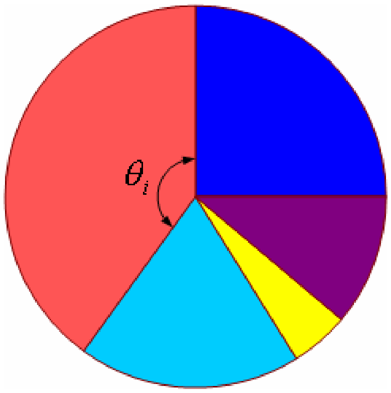


Figure 4.3 A fan-shaped superpixel with its mixture composition.

Similar to the mapping in Layer I, Layer II maps a superpixel center to the corresponding vertex and the radius of a superpixel is $u/2$ as shown in Figure 4.4. The space in the pie chart will be filled by the color of the endmember that has been assigned to this wedge. The color of the points that are not covered by the superpixels is set to black, i.e., **c**=[0, 0, 0]^T. A color in Layer II is referred to as C_{II}.

4.4 Blending

After colors have been assigned to the two layers, Layer I, which gives the general distribution, and Layer II, which gives the details of the distribution, the final display can be generated. In order to display the two layers on the same device simultaneously, the color in the final image is determined by blending C_I and C_{II}. The blending is achieved by linearly combining C_I and C_{II}, which can be described as:

$$C_{final} = t\,C_I + (1-t)\,C_{II} \text{ for } 0 \le t \le 1. \qquad (4.10)$$

Therefore, the color of the final image is within $[C_I, C_{II}]$. When *t,* the blending parameter, approaches 1, the final color trends to C_I, which means the final image displays the general distribution information; when *t* is close to 0, the final color trends to C_{II}, which means the final color image gives the detail distribution information.

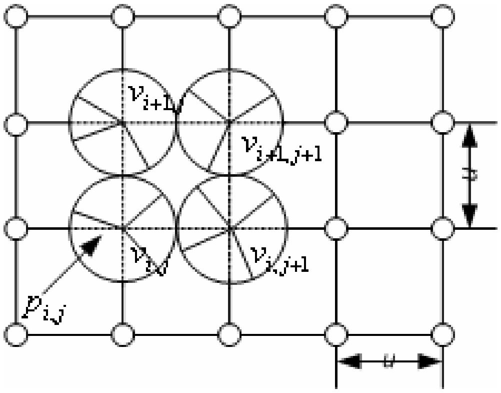
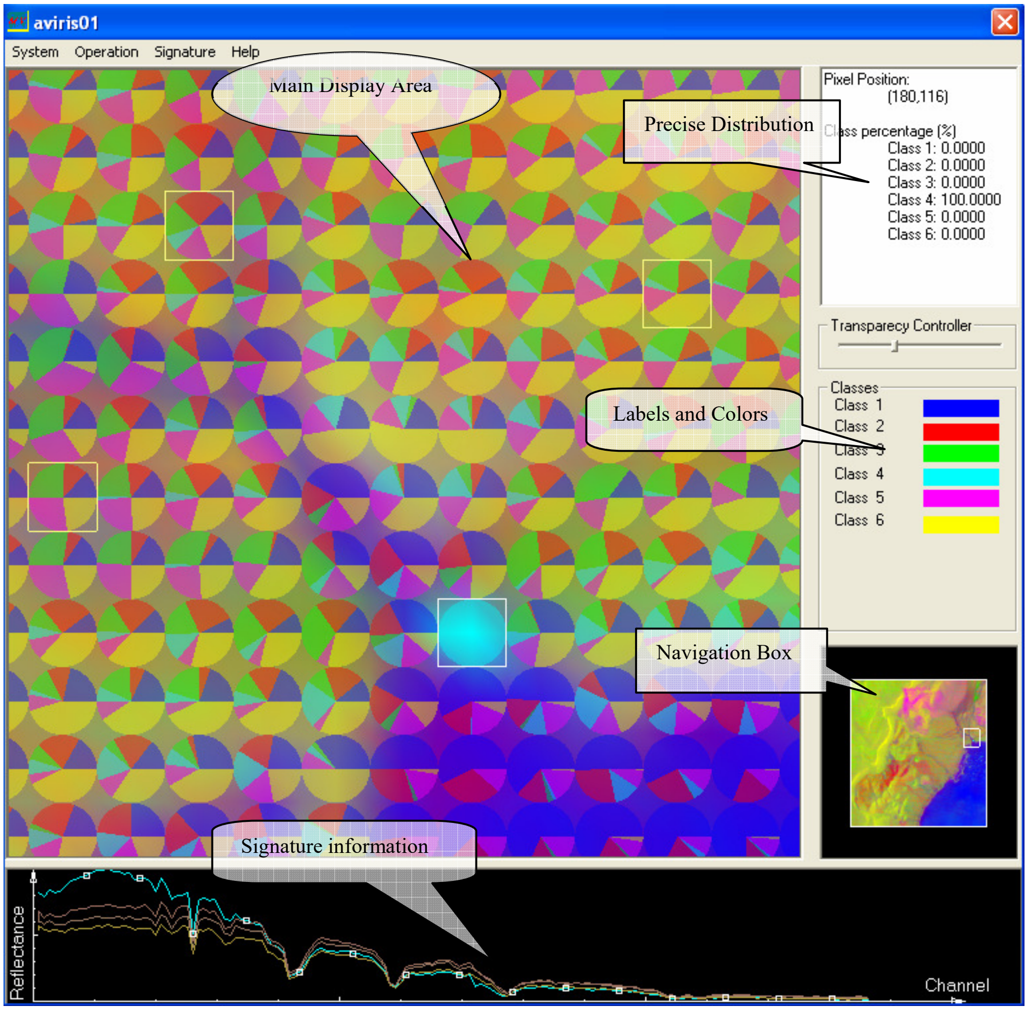


Figure 4.4 The mapping from pixels to superpixels.

4.5 Visualization Exploration Tools

Visualization research has shown that interaction is an effective way to help viewers analyze data. In order to help dig into the details of endmember distribution, the developed visualization system provides several interactive exploration tools to provide additional information in the main visual interface. The main visual interface as shown in Figure 4.5 is divided into five linked view areas: main display area, navigation display area, precise pixel information display area, signature information display area, and class labels and colors.



Pixel Position:
(180,116)

Class percentage (%)
Class 1: 0.0000
Class 2: 0.0000
Class 3: 0.0000
Class 4: 100.0000
Class 5: 0.0000
Class 6: 0.0000

Figure 4.5 The interface in the DBLY visualization system.

In the main display area, viewers can easily manipulate the visual result, such as zooming-in, zooming-out, and panning to select the region of interest (ROI) area. By selecting an ROI, one can obtain the desired degree of details. For example, by selecting a large area, Layer I will dominate the display, whereas the mixing details of each pixel are more visible if a small ROI is chosen. If a very small area is selected, the precise quantitative mixing information of each individual pixel can be seen. To avoid losing the spatial position when viewers select a very small ROI, a small navigation box is displayed to indicate the position of the ROI. Also, it can be used to change the ROI.

Although the distribution details can be revealed by Layer II, it may still be difficult for viewers to figure out the very precise distribution of endmembers. Therefore, the precise pixel information can be displayed, which includes the pixel position and exact abundances. An endmember's signature is a unique reflectance value in spectrum. Two endmembers that are indistinguishable in one spectral range may be very different in another portion of the spectrum. Therefore, signature is an essential property for identifying different endmembers in data analysis. Viewers can choose the spectral signatures to be shown and the color of the signature is the same as the pixels in Layer I, which is also presented in the class color label area. Other information can be easily added to the visual interface.

4.6 Visualization Results

Three examples demonstrate the performance of the developed visualization system in this section. They represent three different cases: the AVIRIS Lunar Lake data does not have noisy pixels, the AVIRIS Low Altitude data is contaminated by isolated noisy pixels, and the Hyperion data contains lines that are noise. The two AVIRIS data are free downloads from the AVIRIS website (aviris.jpl.nasa.gov). The AVIRIS Lunar Lake data is the area of the Lunar Crater Volcanic Field in Northern Nye County, Nevada. The Hyperion data was obtained from the Earth Resources Observation Systems

(EROS) Data Center of United States Geological Survey (USGS) and is an area south of Mississippi State University.

The two color assignment strategies were used for these three examples. Because we have some prior knowledge about the AVIRIS Lunar Lake scene, the vegetation was pre-assigned to green. The rest of the class colors were automatically assigned based on signature similarity. The second and third datasets implemented automatic color assignment with the signature similarity constraint only, since no prior knowledge of endmember types was available.

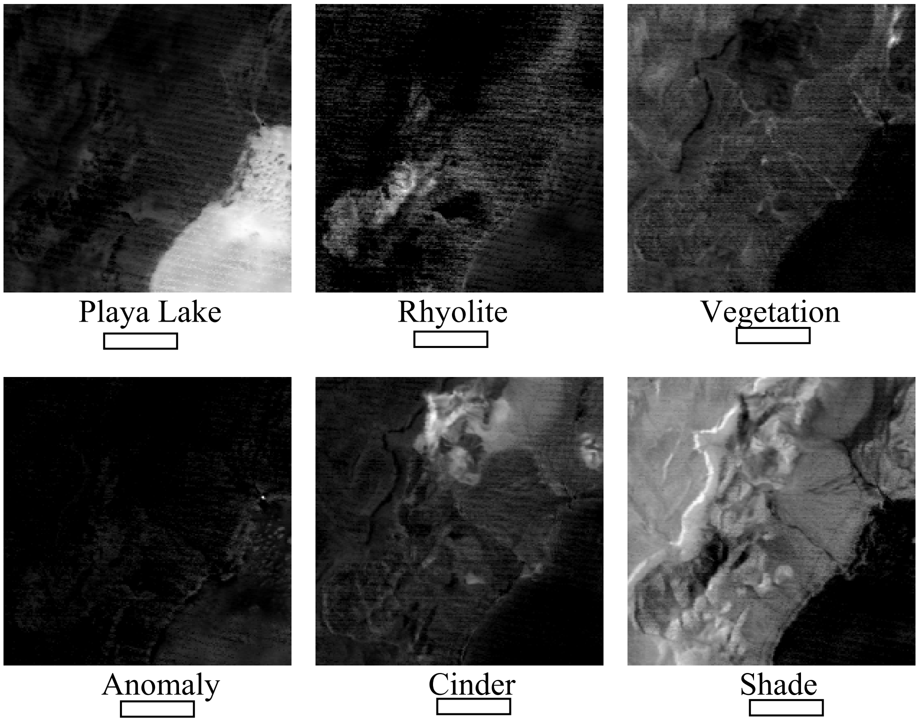


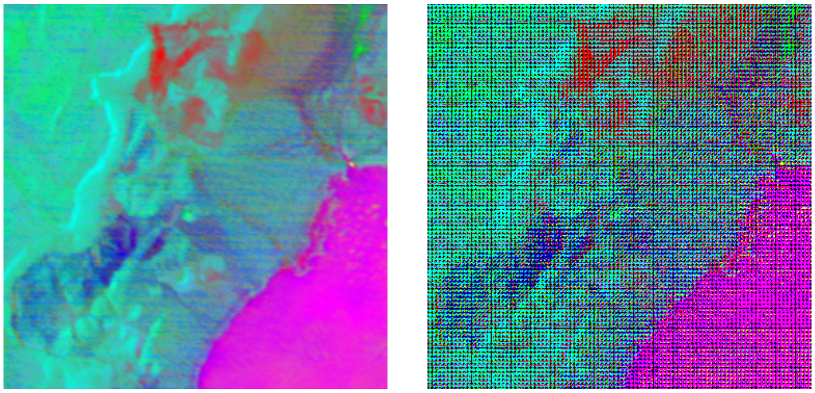
Figure 4.6 The abundance images of the AVIRIS Lunar Lake and the assigned colors.

4.6.1 Sample I: AVIRIS Lunar Lake

The AVIRIS Lunar Lake data shown in Figure 3.4 was visualized by the DBLY algorithm. Figure 4.6 shows the gray-scale images of the six material abundances and their color labels.

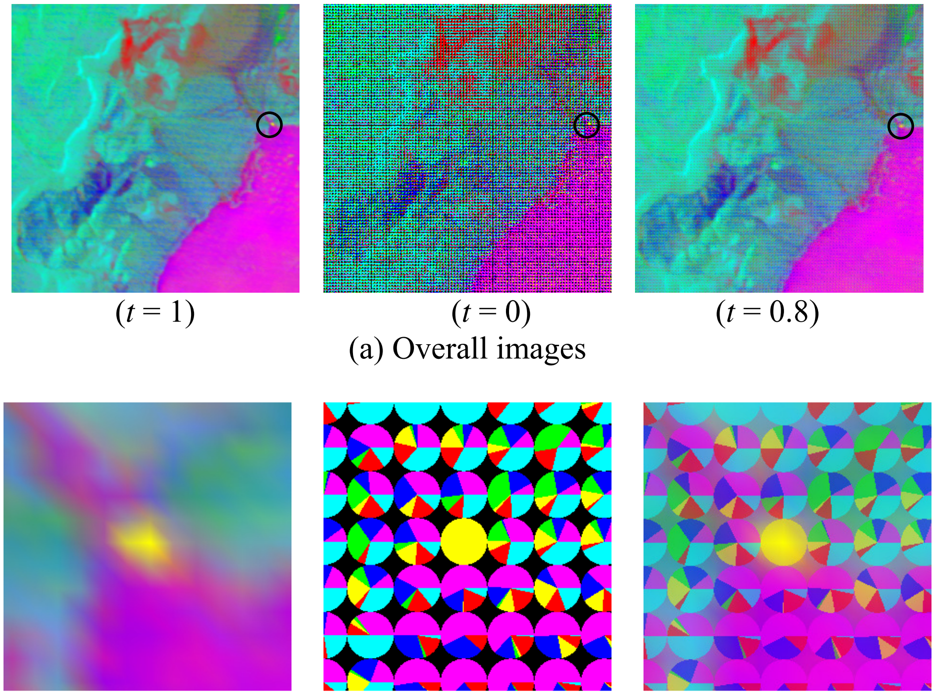
Figure 4.7 shows the Layer I and II images, in which each material was assigned the color indicated in Figure 4.6. The rightmost abundance image in the first row of Figure 4.6 is vegetation, so it is displayed in green. In Figure 4.7(a), the six endmembers can be easily distinguished by color, such as magenta, blue, green, yellow, red, and cyan. In particular, an anomaly highlighted in the circle is displayed in yellow. Such a color display shows the overall material distribution and the fuzzy memberships of a pixel, but it cannot represent precisely the percentage of materials within each pixel. By introducing a second layer as shown in Figure 4.7(b), more detailed information can be revealed.

More details about the final color composite are shown in Figure 4.8. The first column is the result when only Layer I was used to visualize the six endmembers and the second column was generated when only Layer II was used. The third column is the result of blending the two layers with the specified values of *t*. On the other hand, the first row in Figure 4.8(a) shows the complete image using different values of *t*, while the second row in Figure 4.8(b) is an ROI, where individual superpixel disks are easily distinguished. Each disk represents a pixel, and the pie-chart indicates the abundance of each material. In Figure 4.8(b), we can easily see the anomaly pixel and its composition in the context of the neighborhood. This important information cannot be conveyed by viewing the original gray-scale abundance images one after the other.



(a) Layer I (b) Layer II

Figure 4.7 The composite image based on the color assigned in Figure 4.6.



(t = 1) (t = 0) (t = 0.8)

(a) Overall images

(t = 1) (t = 0) (t = 0.5)

(b) Details of pixels (the anomaly and its neighboring pixels)

Figure 4.8 Visualization results for the AVIRIS Lunar Lake data (from left to right: Layer I, Layer II, and DBLY approach).

4.6.2 Sample II: AVIRIS Low Altitude

An AVIRIS Low Altitude subimage with size 512 × 512 as shown in Figure 4.9 was preprocessed before visualization. 132 bands were kept after water absorption and low SNR bands were removed. Isolated noisy pixels were removed using the algorithm in chapter III. Without prior knowledge about this image scene, thirty materials were generated first and seven meaningful material signatures were kept after abundance image selection and similar signature combination. Then seven abundance images were produced by applying the supervised FCLSLU. Figure 4.10 displays these images and the automatically assigned colors.

Figure 4.11 shows the final visualized images, such as Layer I (first column), Layer II (second column), and DBLY (third column). The blending parameter *t* controls the amount of Layer I and Layer II in the final color display. The second row in Figure 4.11 is an ROI, a yellow spot, which is located in the white circle in the first row. Without using the DBLY display, viewers may be able to tell this position contains some "yellow" material, but cannot gain a clear understanding of its distribution. Our visualization system shows the yellow spot not only contains this "yellow" material but also three other materials, and this yellow material is the major component. The viewer can easily know the percentage of each material by looking at the top-right window in the graphic user interface (GUI) as shown in Figure 4.5.

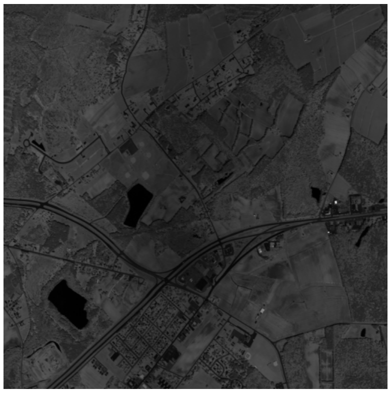


Figure 4.9 The AVIRIS Low Altitude scene of size 512 × 512.

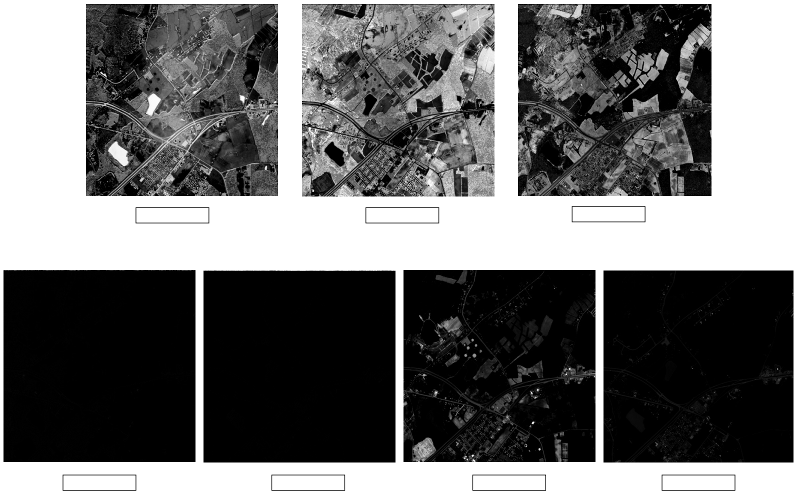
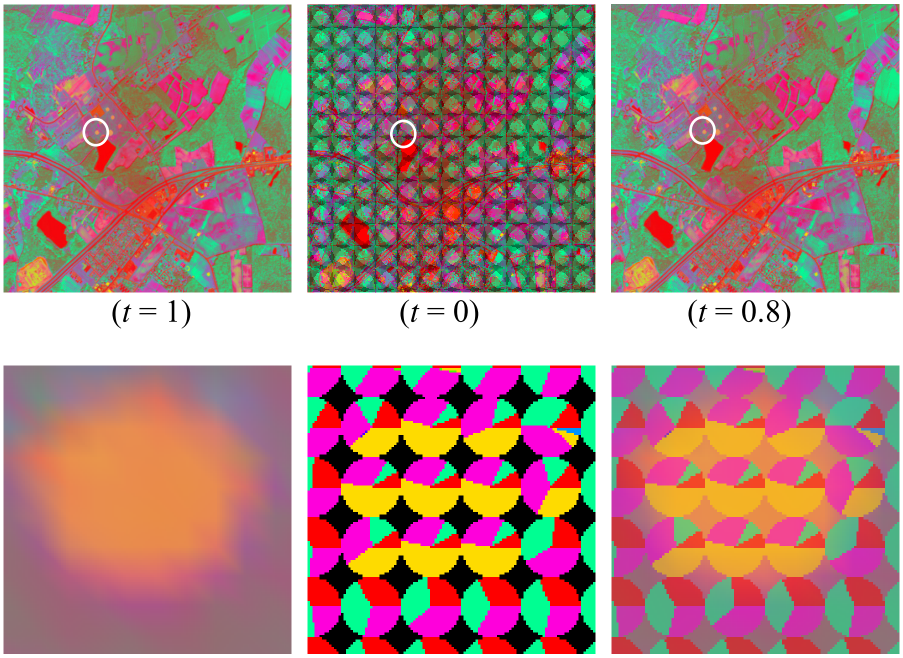


Figure 4.10 The abundance images of AVIRIS Low Altitude and their colors.



(t = 1) (t = 0) (t = 0.8)

(t = 1) (t = 0) (t = 0.5)

Figure 4.11 Visualization results for the AVIRIS Low Altitude data (from left to right: Layer I, Layer II, and DBLY).

4.6.3 Sample III: Hyperion Experiment

The Hyperion data shown in Figure 4.12 was used to test the system as well. After water absorption and low SNR bands were removed, 152 bands remained. As the first spaceborne hyperspectral sensor, the images it produces contain lots of sensor noise such as dark lines. These dark lines greatly affect the classification results, so the algorithm in Chapter III was employed to detect and remove these dark lines. After the dark lines were removed, twenty materials were extracted using the unsupervised FCLSLU. Then five signatures were kept after the selection of meaningful abundance images and combination of similar signatures. The five abundance images generated by the supervised FCLSLU and the automatically assigned colors are shown in Figure 4.13.

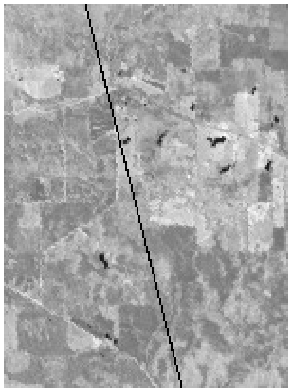


Figure 4.12 A Hyperion image scene of size 150 × 200.

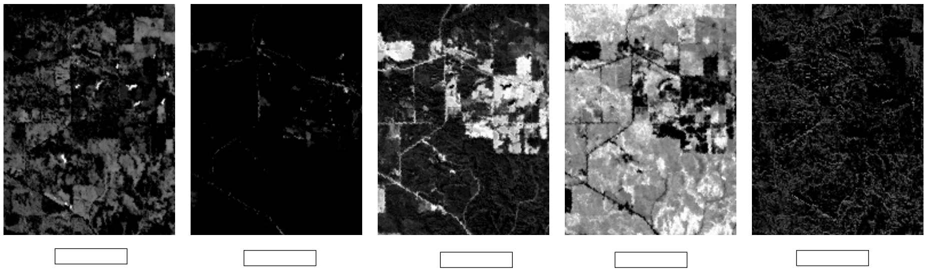
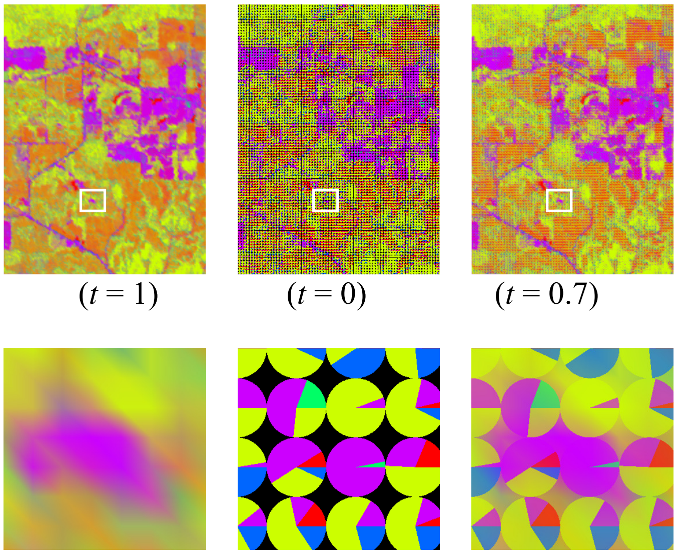


Figure 4.13 Five abundance images of the Hyperion data and their colors.



(*t* = 1) (*t* = 0) (*t* = 0.7)

(*t* = 1.0) (*t* = 0.0) (*t* = 0.4)

Figure 4.14 Visualization results for the Hyperion data (from left to right: Layer I, Layer II, and DBLY).

Figure 4.14 shows the color visualization comparison among the Layer I, Layer II, and DBLY. For example, the distribution of the material in purple corresponding to the buildings and roads is easily seen in the first row; the small purple area highlighted in the first row is displayed as the ROI in the second row where we can see that none of the pixels are pure and the largest abundance is about 90%. This experiment further demonstrates that using DBLY we can display both the general and detailed information as needed.

4.7 Conclusion

This chapter presents the approach of visualizing hyperspectral images by employing double color layers. It is an improvement from the conventional gray-scale display that needs to show several images one after another, and it also improves the results from other methods that display the general distribution of endmembers only. The DBLY visualization technique can simultaneously display the overall endmember spatial distribution and their composition at the subpixel level.

In the following chapters, we validate the DBLY scheme by employing quantitative evaluation via user study, and compare it with existing algorithms.

CHAPTER V

AN EMPERICAL VALIDATION OF DBLY VISUALIZATION ON SYNTHETIC DATASETS

A user study can not only evaluate the effectiveness and weaknesses of visualization techniques, but also provide new findings and guidelines for further improvement. In this dissertation, we employ user studies to investigate the efficiency of the DBLY visualization technique. We mainly focus on its information conveying capability from both global and local aspects. We compare it with the existing algorithms: gray-scale side-by-side display (denoted as GRAY), colored hard classification (denoted as HARD), and colored soft classification (denoted as SOFT). In this chapter, we will describe the design, results, and findings of the study based on synthetic datasets. In Chapter VI, we will present a user study using a real hyperspectral dataset.

5.1 Experimental Design and Settings

5.1.1 Domain Questions

Shippert stated that a project utilizing hyperspectral imagery usually has the purposes of target detection, material mapping, material identification, and/or mapping details of surface properties [12]. In military applications, hyperspectral imagery has been used to detect military vehicles covered by vegetation canopy or to discriminate true objects from fake ones. Cochrane successfully identified vegetation species [77], and Coops *et al*. accessed vegetation stress and disease from hyperspectral images [78]. In general, the practical usage of hyperspectral imagery can be described by one or more of the following generic and domain questions:

- *Classification*: What and how many endmembers are present in an image scene? (Practical domain questions may be "what are the different kinds of land use patterns present in this image?" or "is there any hidden military vehicle?")

- *Relative Position*: Where are those endmembers relative to each other? ("Where is the wheat field infested with bugs?" or "how far are the weeds from residential areas?")

- *Perceptual Edge*: How widely distributed are the endmembers in a region? ("Where is the edge of the contaminated water body?" or "how wide does the wheat disease spread?")

- *Quantification*: How much of an endmember is in a small region or the whole area? ("How many bugs are in this small area of the wheat field?" or "how serious is the stress in a citrus farm?")

In many cases, how useful a hyperspectral image is depends on how well these questions can be answered. The goals of this study are to investigate the performance of the four chosen visualization methods in answering these questions. However, designing a user study to explicitly test these questions may not be feasible because exploring a dataset is a complex cognitive behavior. After consulting with remote sensing experts, we decided to investigate two important aspects of understanding hyperspectral images in this research: global pattern display and local information conveyance. We have three hypotheses in our user studies:

H1: Displaying materials in one image is better than displaying them in several side-by-side images when trying to ascertain relative location of materials.

H2: Adding a pie-chart layer does not degrade the background layer (SOFT)'s capability in displaying global patterns.

H3: The Pie-chart layer increases DBLY's efficiency in revealing local information.

Based on the three hypotheses, we designed five study tasks, where three test the capability of global pattern display and two test the ability of local information display.

5.1.2 Datasets and Colors

Because of the availability of ground truth, synthetic datasets are widely used in measuring visual perception. Ward *et al*. addressed several advantages of using synthetic rather than real datasets [56]. Thus, we used synthetic rather than real datasets in the pilot study.

Since the GRAY technique displays each endmember as a separate image, it is not easy for a user to switch from one image to another image quickly to find answers if we display GRAY as four full screen images. We found that four endmembers were an effective trade-off for this user study because we can maximize usage of the screen space by displaying the grayscale images in a 2 × 2 square. The size of each gray image is 400 × 400. The other visualization techniques display the image as 800 × 800. The size of datasets that were used to test the capability of global pattern display is 200 × 200. 20 × 20 pixel blocks of the datasets were used to simulate the zooming-in function to validate the ability of conveying local information.

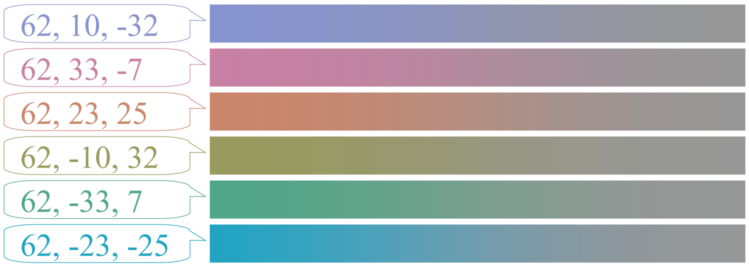


Figure 5.1 Color ramps used for all the experiments. Each set of three parameters is the value of the left end of a color ramp in the $L^*a^*b^*$ color space; the right end of each is (62, 0, 0).

Color is an important component in the selected techniques. Although we designed synthetic datasets that only contained four endmembers, we used six colors in order to study more complex color combinations. We used the six colors employed by Hagh-Shenas *et al.* [58] from $L^*a^*b^*$ color space in this study, but we kept the luminance constant. The color ramps are presented in Figure 5.1. Since all of the synthetic datasets only contain four endmembers, we randomly chose four colors from the six possible colors to represent each endmember.

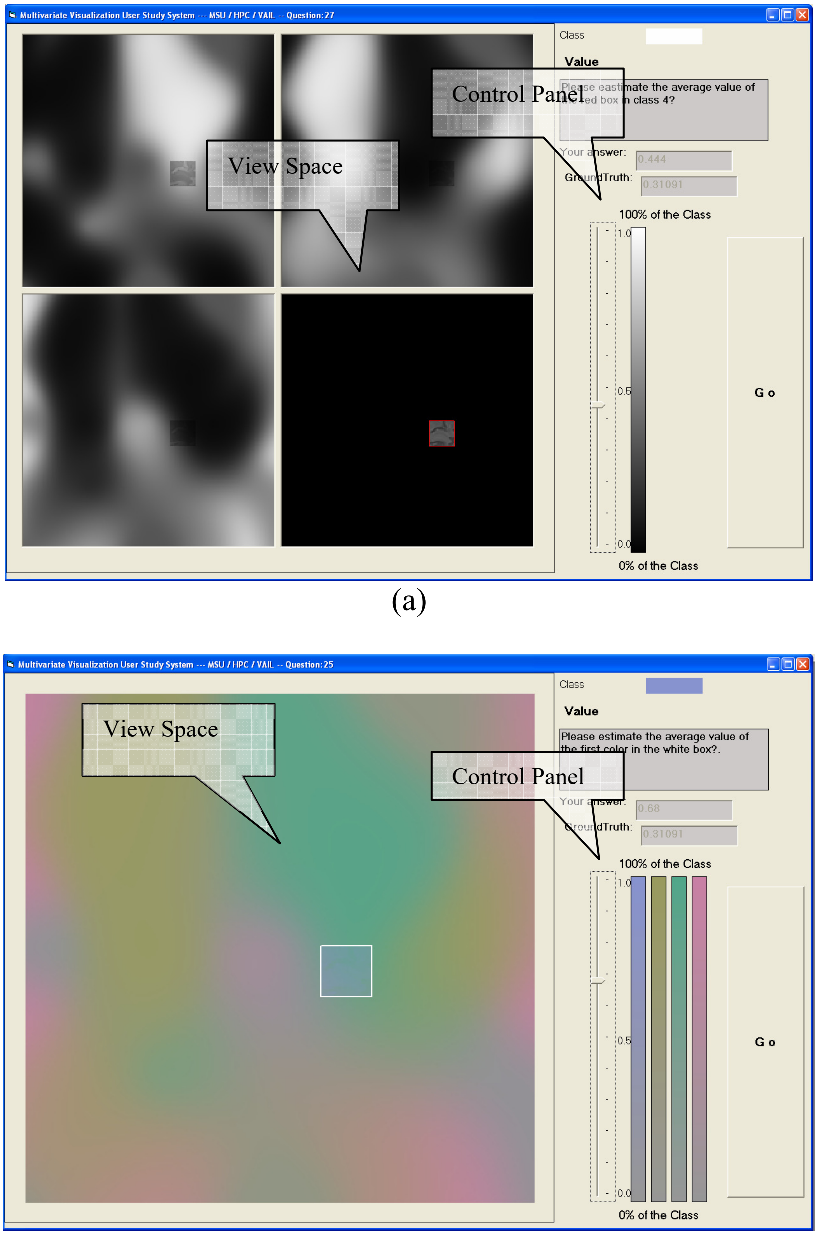
5.1.3 GUI and Software

To implement the study, we developed a GUI for participants to perform the tasks; several scripts were written to generate the synthetic datasets and analyze the results. The GUI is displayed in Figure 5.2. All of the test images were displayed in the view space and an external mouse was used to respond either within the control panel or directly onto the view space. For the GRAY images, the view space was split into a 2 × 2 matrix as shown in Figure 5.2(a). If the test image had chrominance (HARD, SOFT, and DBLY), the image was displayed at the center of view space as in Figure 5.2(b). For each participant, a list of pre-determined tests was displayed by the GUI. Their answer and the time required to complete each test were recorded by the GUI and written to an output file for further analysis.

The test list was generated programmatically. First, a sequential list of tests was generated by looping through all of the tasks, visualization techniques, and all the datasets. Then, the sequential list was randomly permuted to control for learning and fatigue effects. Finally, the permuted test list was written to a formatted file with other testing information, such as techniques and ground truth answers. The formatted file was loaded by the GUI for training and real testing. The same tests were presented to all participants, but the test order was randomized per participant.

5.1.4 Participants

At the beginning of this study, we planned to recruit participants with backgrounds in remote sensing. However, hyperspectral imagery is widely used in many applications and is not limited to remote sensing. Also, our tasks have a low cognitive level and do not require a strong background in remote sensing to complete. Ten university graduate students performed this study. Six of the participants were in remote sensing, two in visualization, one in computational fluid dynamics, and one in agriculture economics.


Multivariate Visualization User Study System --- MSU / HPC / VAIL -- Question:27
View Space
Control Panel
Class
Value
100% of the Class
1.0
0.5
0.0
0% of the Class
G o

Please eastimate the average value of the red box in class 4?

Your answer: 0.444

GroundTruth: 0.31091

(a)

Multivariate Visualization User Study System --- MSU / HPC / VAIL -- Question:25
View Space
Control Panel
Class
Value
100% of the Class
1.0
0.5
0.0
0% of the Class
G o

Please estimate the average value of the first color in the white box?.

Your answer: 0.68

GroundTruth: 0.31091

(b)

Figure 5.2 Samples of the GUI in training mode in pilot study. (a) The GUI with four gray-scale images; (b) the GUI with one SOFT image which is generated from the same dataset shown in (a).

We administered the test with a laptop computer, with its display profile set to the standard RGB color space. We removed distracting icons and screen clutter from the screen space. The study was run in a conference room with only the experimenters and the participant present. Before the study, participant completed necessary paperwork and answered a general questionnaire. We paid participants $10 for participating in the study

To reduce any potential training bias, we wrote a training guide so that all participants received the same training. We used similar but different datasets for training and testing. Training consisted of 20 tests, which covered all 4 techniques and 5 tasks. During training, the experimenter walked the participant through each test in sequence. After this, the experimenter asked the participant to complete the same 20 tests and following random sequence tests. However, we did not require participants to finish all the training tests; we allowed them to start the experiment early if they desired. We encouraged participants to ask questions during the training. Testing lasted only 40 minutes to avoid fatigue effects. Participants were not required to finish all the tests. The entire study took about an hour, including the paperwork and training.

5.1.5 Independent and dependent variables

Table 5.1 lists the independent and dependent variables measured in this user study. The independent variables included *participant, technique, and the tuple (task × dataset)*. As specified in Table 5.1, each task was built upon either 9 or 10 datasets; there were a total of 48 *task* × *dataset tuples*. The experiment consisted of 4 *techniques* × 48 *tuples* = 192 possible tests.

The quantified dependent variables were *response time, relative error, absolute error, and normalized error*. Standard error plots and one-way analysis of variance (ANOVA) was employed to analyze dependent variables. The detailed analysis is discussed in the following section.

Table 5.1

THE INDEPENDET VARIABLES AND DEPENDETN VARIABLES STUDIED IN THE POLIT STUDY.

INDEPENDENT VARIABLES		
participant	10	(random variable)
technique	4	GRAY (grayscale side by side) HARD (colored hard classification) SOFT (colored soft classification) DBLY (double-layer)
task × *num datasets*	48	relative location × 10 perceptual edge location × 10 block value estimation × 9 class recognition × 10 target value estimation × 9
DEPENDENT VARIABLES		
response time	measured for each test in seconds	
relative error	*user answer – ground truth*	
absolute error	| *relative error* |	
normalized error	$\frac{user\ answer - ground\ truth}{bar\ length} \times 100\%.$ (measured for perceptual edge detection task)	

5.2 Experimental Tasks and Results

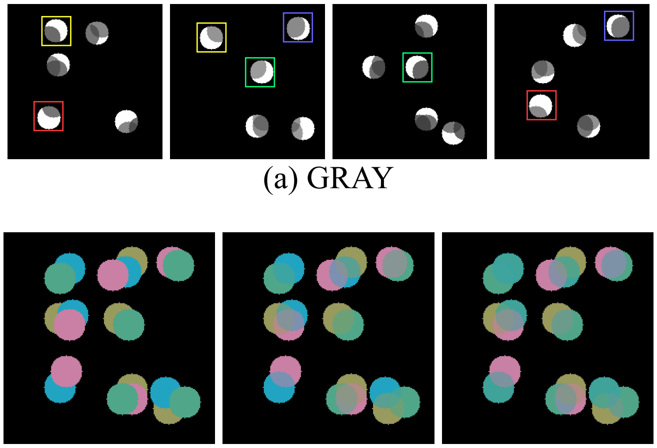
In this section, we present the five tasks and discuss the results for each task. ANOVA was used to analyze the data. For the ANOVA, we modeled our experiment as a repeated-measures design that considers participant a random variable and all other independent variables as fixed. The ANOVA enables us to analyze the difference of the means among the four tested visualization techniques. The *F*-value and *p*-value are generated by the ANOVA. A large *F*-value ($F>>1$) indicates that the visualization techniques have a major effect on the mean of the measured variables and the lower the *p*-value, the more significant the difference of means among the measured variables for the tested visualization techniques. However, the ANOVA does not specifically indicate which pair of visualization techniques exhibits statistical difference. Post-hoc tests can be applied to determine which specific pair(s) of visualization techniques is differentially expressed. We utilized the Ryan procedure (REGWQ) for post-hoc multiple comparisons [59]. We processed outliers in the data with the procedure described by Barnett *et al.* [60]. We determined outliers by examining the tails of the distributions and noting values that appeared after conspicuous gaps in the histogram. Each outlier was replaced by the median of the remaining values in the experimental cell. Given that outliers are considered mistaken values, this procedure improves the calculation of means, standard errors, and the sums-of-squares terms used in ANOVA, which would otherwise be inappropriately influenced by the outlying values.

5.2.1 Global Pattern Display Capability

Patterns reveal important information to help humans understand the real world. One of the major advantages of visualization is that it can display the pattern directly to participants or give participants an opportunity to recognize patterns very quickly. There are many properties of a pattern that can be tested. This study investigates three aspects of the pattern display capability of the visualization techniques: relative location of classes, perceptual edge detection, and block value estimation.

Relative Location

The goal of this task was to quantify the ability of each visualization technique to represent the relative location of variables.



(a) GRAY

(b) HARD (c) SOFT (d) DBLY

Figure 5.3 A sample set of images for testing relative location with 4 cases of 2-circle overlap. For the GRAY visualization, similarly-colored boxes indicate matching 2-endmember sets; there are a total of 4 such sets for this example. During the actual experiment, the colored boxes were not shown.

Task: The datasets which were used in this task consisted of four co-located endmembers. Each endmember was presented by five solid circles. The location of the circles was determined by the following steps:

- Evenly dividing the image scene into 3 × 3 blocks
- Circles were randomly put into the center of blocks.
- Small offset was added to each circle.

A circle in one endmember may overlap with circle(s) in other endmembers. The overlapped circles were used to simulate co-existing real-world materials. In this task, we asked participants to indicate the number of cases where exactly two endmembers overlapped. Participants completed a forced-choice decision by pressing one of 7 buttons, which indicated 0 to 6 cases. This was visually apparent when two circles overlapped. Since the overlaps of circles were determined by the relative locations of the circles, this task can be used to test the capability of visualizing relative location.

One of the sample dataset is displayed in Figure 5.3, where the correct answer is "4". Figure 5.3(a) is the GRAY visualization, where the white parts contain 100% of that endmember and no other endmember exists in the white parts. But there are more than one endmember that co-exist at the gray parts of circles.

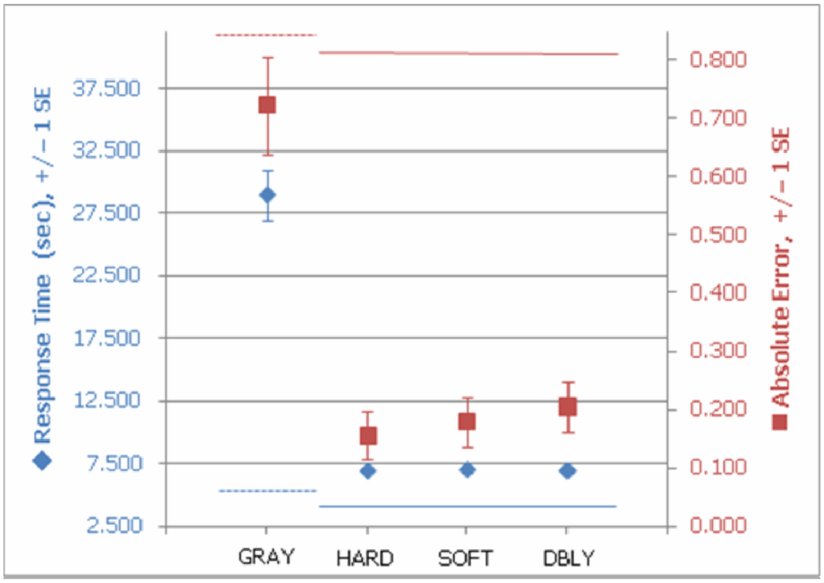
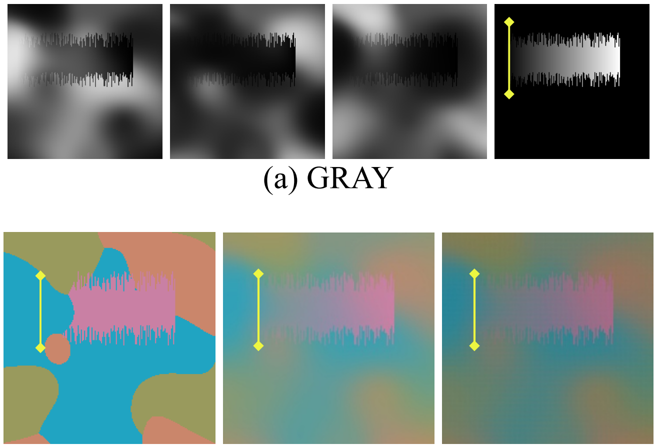


Figure 5.4 Results from locating the relative position for both dependent measures. For this and all figures, absent error bars indicate the standard error is smaller than the symbol size. The color-coded lines indicate the result of post-hoc comparisons.

Results: The dependent measures were *response time* in seconds, and *absolute error* (Table 5.1). The *absolute error* ranged from 0 (no error) to 5 (maximum error). We recorded a total of 352 answers, including 90 answers for GRAY and HARD, 84 for SOFT, and 88 for DBLY. Of the 352 answers, 94 contained an *absolute error* > 0; in practice the recorded error ranged from 1 to 3. The means are displayed in Figure 5.4. We found a major effect of visualization technique on both *absolute error* ($F(3, 27) = 12.59$, $p < .000$) and *response time* ($F(3, 27) = 42.44$, $p < .000$). As shown in Figure 5.4, we used the color-coded lines to represent the REGWQ post-hoc comparison results for both measurements (i.e., the blue lines for *response time*, and the red lines for *absolute error*). The different line styles represented the different groups in the results of the post-hoc comparison. In this task, the post-hoc comparisons indicated that GRAY fell into one group, and HARD, SOFT, and DBLY fell into the other group in both measurements. This analysis shows that GRAY resulted in lower performance than the other techniques, and that the performance of HARD, SOFT, and DBLY was at the same level.



(a) GRAY

(b) HARD (c) SOFT (d) DBLY

Figure 5.5 An example of the perceptual edge detection task. The yellow lines indicate the ground truth positions.

Perceptual Edge Detection

Here we define the perceptual edge as the position where a color can no longer be perceptually distinguished from the background. The perceptual edge is important because some endmembers may no longer be present at a certain position. This task was designed to test how well each visualization technique preserves the perceptual edge.

Task: Three endmembers were generated by the Matlab interpolation function *griddata* to simulate a continuous distribution of materials over the image scene. We represented the fourth endmember by a bar whose value varied right-to-left from 1.0 to 0.0. We set the bar length to 150 pixels, and randomized the starting position within the square [0, 50] × [50, 150]. We added a small offset to each vertical column to reduce the otherwise sharp horizontal edges. The sample images are shown in Figure 5.5. For the GRAY visualization, we used the constraints of sum-to-one to determine the values at matching pixels. For this task we asked the participants to click on the left perceptual edge of the fourth endmember.

Results: We recorded the x-coordinate of the user's mouse click for this task. The dependent measures were response time and normalized error. A total of 351 answers were recorded, including 91 for GRAY, 88 for HARD, 83 for SOFT, and 89 for DBLY. We eliminated 13 normalized error outliers. The means are displayed in Figure 5.6. We found a major effect of visualization technique on normalized error ($F(3, 27) = 124$, $p < .000$), but no effect of response time ($F(3, 27) = 1.07$, $p = .375$). The post-hoc comparisons indicated the differences shown in Figure 5.6. Although the post-hoc comparisons indicated a difference in response time and normalized error between

GRAY, SOFT and DBLY, the normalized error is less than 5% (shown in Figure 5.6). The normalized error of HARD is far larger at 28.8%. The high normalized error of HARD comes from the hard classification itself. Since there are four endmembers in the study, value less than 1/*p* (*p* is the number of the endmembers) will not be the largest value.

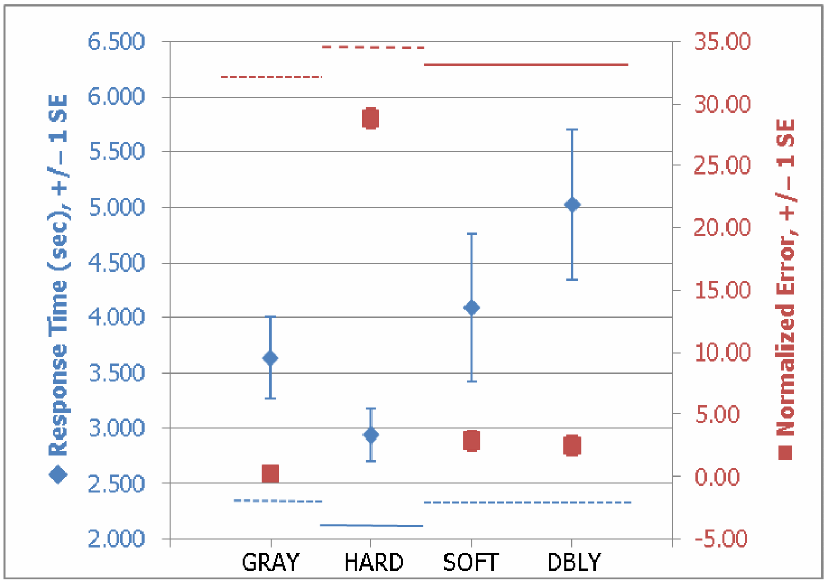
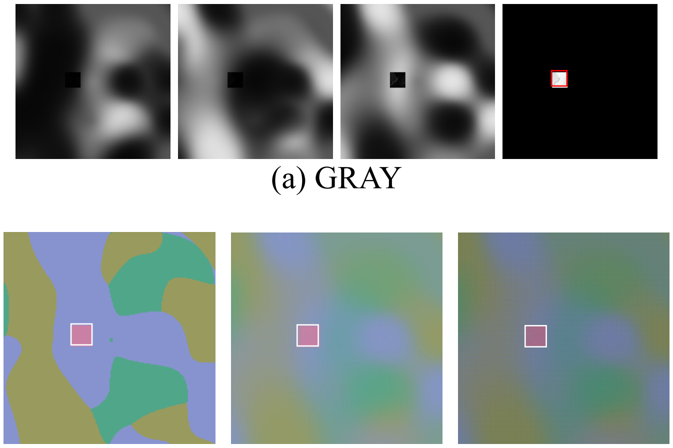


Figure 5.6 Result of the perceptual edge detection task. The lines indicate the results of post-hoc comparisons.

Block Value Estimation

The block value estimation task was designed to assess participants' ability to accurately determine the continuous value encoded by a color range. In each region, colors represent overlapped multiple scalars.

Task: As with the perceptual edge detection task, three endmembers were generated by the Matlab interpolation function *griddata* to simulate a continuous distribution of materials over the image scene. The fourth endmember is represented by a 20 × 20 pixel block. Unlike Hagh-Shenas *et al.* [58], where the color in the tested region was constant, the color in the block varies in the range [*low, low* + 0.2], where *low* varies in the range [0.0, 0.1, 0.2, …, 0.8]. The task was to match the block color using a slider that indicates a choice from the adjacent endmember color ramp (see Figure 5.2(b)). A sample dataset is displayed in Figure 5.7. In the sample images, the block in the fourth endmember is colored by white in the GRAY image, while it is colored by pink in other techniques. In Figure 5.7, the maximum value in each block is 1.0, the minimum value is 0.80, and the average value is 0.93.


(a) GRAY

(b) HARD (c) SOFT (d) DBLY

Figure 5.7 A sample of the block value estimation task. Figure 5.2 shows the GUI for this task.

Results: The dependent measures were *response time* and *absolute error*. The *absolute error* ranged from 0 (no error) to 1.0 (maximum error). We recorded a total of 308 answers, including 79 answers for GRAY, 73 for HARD, 77 for SOFT, and 79 for DBLY. We eliminated 12 *absolute error* outliers; after doing so the maximum *absolute error* for any participant was 0.46. To conduct the analysis for this task, we had to remove the data for one participant, because that participant did not complete any DBLY tasks. We found a major effect of visualization technique on *absolute error* ($F(3, 24) = 11.05, p < .000$), but no effect of *response time* ($F(3, 24) = .96, p = .354$). The post-hoc comparisons indicated the differences shown in Figure 5.8; the HARD technique was significantly worse than the other techniques.

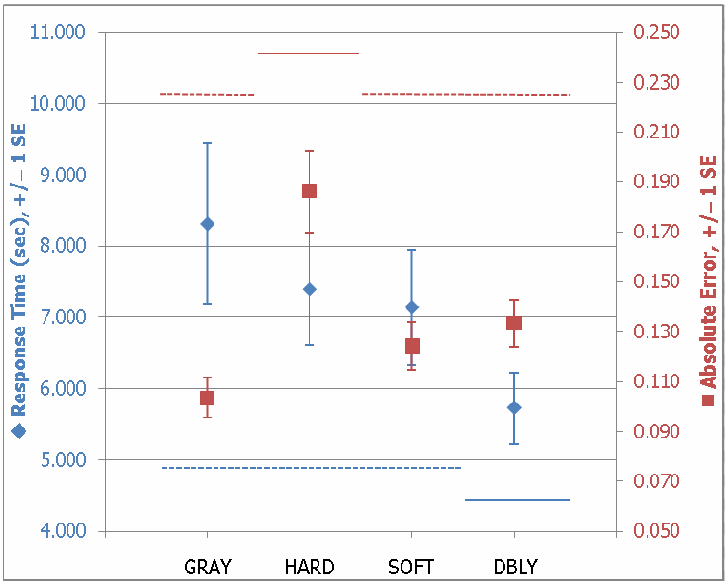


Figure 5.8 Result of the block value estimation task. The lines indicate the results of post-hoc comparisons.

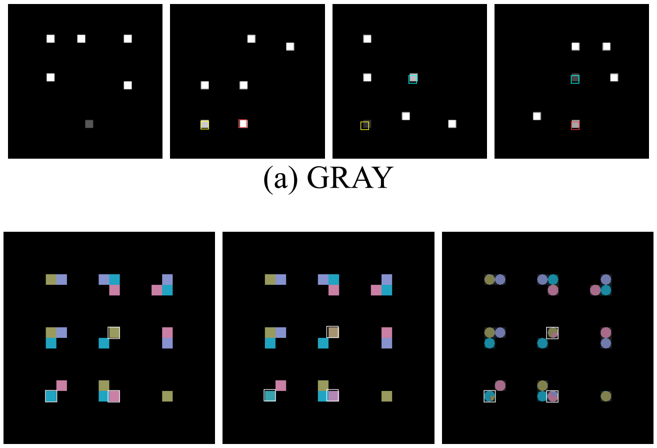
5.2.2 *Local Information Conveying Capability*

Compared to its high spectral resolution, the spatial resolution of hyperspectral imagery is relatively low. For example, a pixel in the Hyperion dataset represents 30 × 30 square meters. The high spectral resolution of hyperspectral imagery gives researchers the opportunity to classify and analyze landcover at the subpixel level. Interaction is indispensable for investigating the detailed information.

In this section, we designed two tasks to evaluate the capability of visualization techniques in conveying detailed information at the subpixel level. We synthesized 20 × 20 pixel datasets to simulate the zooming-in operation, and set the blending parameter in DBLY to 0.8 to pop out the pie-chart layer.

Class Recognition

Recognizing the endmembers is the first step in understanding the dataset itself. The high spectral resolution of hyperspectral imagery may permit the exploration of detailed information in a small area, such as finding a hidden military target in the woods. This goal of this task was to assess participants' ability to determine the number of the endmembers present when zooming into the images.



(a) GRAY

(b) HARD (c) SOFT (d) DBLY

Figure 5.9 A sample set of images for testing class recognition with 3 cases of 2-endmember subpixels. For the GRAY visualization, similarly colored boxes indicate matching 2-endmember pixels; for other techniques, white boxes indicate the 2-endmember pixels. During the actual experiment, the colored boxes were not shown.

Task: Since the size of the dataset for this task was 20 × 20 pixels, each pixel is a square with GRAY, HARD, and SOFT, and a square covered by a pie-chart in DBLY. Each pixel may contain one or more materials to simulate the real-world situation where several endmembers co-exist at the same location. In this task, we asked participants to indicate the number of pixels that contain exactly two endmembers. Figure 5.9 displays a sample dataset. In Figure 5.9(a) (the GRAY visualization) the white pixel contains 100% of that endmember; otherwise, other endmembers co-exist in that pixel.

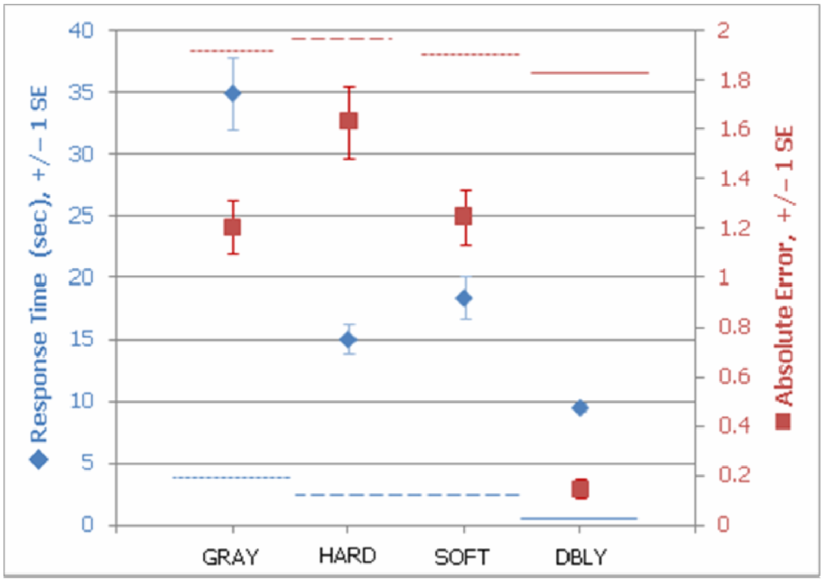


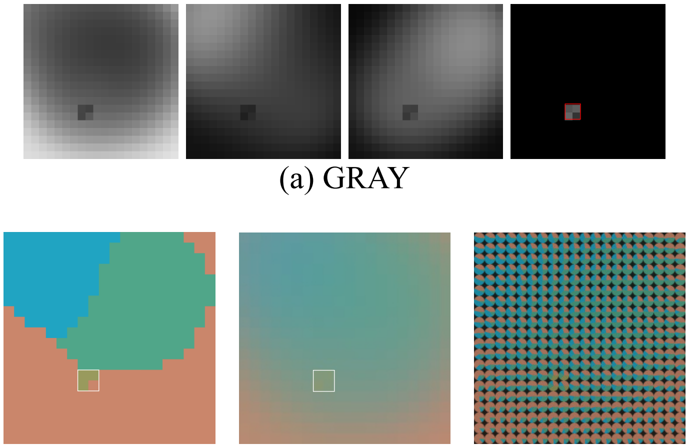
Figure 5.10 Result of the class recognition task for both dependent measures. The color-coded lines indicate the result of post-hoc comparisons.

Results: The dependent measures were *response time* and *absolute error*. The *absolute error* ranged from 0 (no error) to 6 (maximum error). We recorded a total of 361 answers, including 93 answers for GRAY, 89 for HARD, 90 for SOFT, and 89 for DBLY. Of the 361 answers, 218 contained an *absolute error* > 0. The means are displayed in Figure 5.10. We found a major effect of visualization technique on both *absolute error* (*F*(3, 27)

= 43.34, p < .000) and *response time* (F(3, 27) = 13.10, p < .000). The post-hoc comparisons indicated the differences shown in Figure 5.10 for both dependent measures. The results show that DBLY can achieve a very low *absolute error* (0.2) compared to other techniques, and that participants were significantly faster with DBLY as well. With HARD the task is basically impossible, while with SOFT the task is possible but very difficult. Several participants indicated that they resorted to guessing, which explains the relatively low *response time*s for these techniques. With GRAY the task was possible but time-consuming.

Target Value Estimation

This task was designed to evaluate the ability of the four techniques in conveying quantitative information. It is very similar to the block value estimation task. The difference is this task is done on an ROI, not the whole image.



(a) GRAY

(b) HARD (c) SOFT (d) DBLY

Figure 5.11 An example of the target value estimation task. The red/white boxes indicate the target position.

Task: Three endmembers were generated by the Matlab interpolation function *griddata* to simulate a continuous distribution of materials over the image scene. The fourth endmember was the target and only had a value in a 2 × 2 block of pixels. The color in the block varies in the range [*low, low* + 0.2], where *low* varies in the range [0.0, 0.1, 0.2, …, 0.8]. The task was to match the block color using a slider that indicates a choice from the adjacent color ramp (similar to the block value estimation task shown in Figure 5.7). Figure 5.11 displays an example dataset, where the value of the target is in the range [0.2, 0.4], and the average value over the four pixels is 0.32.

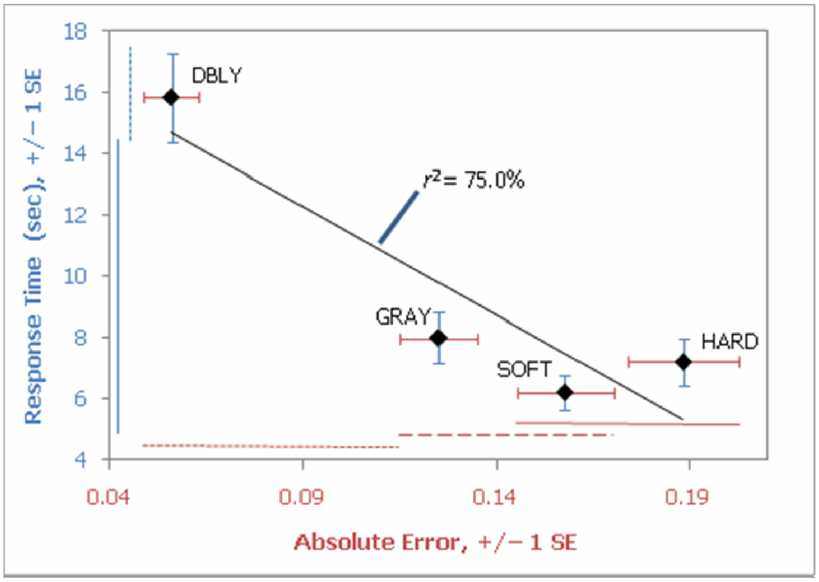


Figure 5.12 Result of estimating the target value. The color-coded lines indicate the result of post-hoc comparisons.

Results: The dependent measures were *response time* and *absolute error*. The *absolute error* ranged from 0 (no error) to .52 (maximum error). We recorded a total of 317 answers, including 79 answers for GRAY, 83 for HARD, 74 for SOFT, and 81 for DBLY. We eliminated 11 *absolute error* outliers. We found a major effect of visualization technique on both *absolute error* ($F(3, 27) = 22.54$, $p < .000$) and *response time* ($F(3, 27) = 10.24$, $p < .000$). The post-hoc comparisons indicated the differences shown in Figure 5.12 for both dependent measures. An overlap of the red lines showed that GRAY and HARD are in different group and that SOFT can not be separated from GRAY and HARD.

A regression line explains $r^2 = 75.0\%$ of the observed variance as shown in Figure 5.12, while a regression line only through GRAY, SOFT, and DBLY explains $r^2 = 95.8\%$. The range of r^2 is [0.0, 1.0]. $r^2 = 100\%$ means that the regression line perfectly fits the means of *response time* and *absolute error* across the visualization techniques and $r^2 = 0.0$ means that there is no linear relationship among the means of *response time* and *absolute error* across the visualization techniques. The regression line shows a clear *response time/accuracy* tradeoff for GRAY, SOFT, and DBLY. Participants were very accurate with DBLY, but it took them longer to study the individual pie charts. GRAY and SOFT require a mental combination of colors. The task was very difficult to perform with HARD, so participants adopted a strategy of answering quickly.

5.3 Discussion

This pilot study indicates that the GRAY approach is not sufficient to locate the relative location and visualize local detailed information; but the GRAY method is effective in displaying the perceptual edge and for participants to estimate block values. Since a pixel in the HARD algorithm is colored by only 1 of *p* colors, where *p* is the number of endmembers, most participants thought the images from the HARD were cleaner than those from any other algorithms. However, our user study found that the

HARD approach is less effective in perceptual edge detection, block value estimation, or local information display.

Although the post-hoc comparisons indicated a difference between the GRAY and SOFT algorithms for perceptual edge detection task, their *normalized errors* were below 5% (shown in Figure 5.6) implying they are comparable for revealing the perceptual edge. The SOFT approach is in the winning group for block value estimation, but it is less efficient in local information display.

The DBLY technique is the most accurate method for local detail conveyance. Taking the advantages from the SOFT method, it is also effective in global pattern display. The user study attests that adding a pie-chart layer to the SOFT approach is necessary for conveying local information while the DBLY algorithm maintains the efficiency in global pattern display that is exhibited by the SOFT method.

In summary, the results of the pilot study indicate that the HARD algorithm is less effective for either global pattern or local information display. The GRAY and SOFT methods are efficient and comparable for showing global patterns, but are less effective for revealing local details. Finally, the DBLY approach is efficient in conveying local detailed information and is as effective as the best traditional methods for global pattern depiction.

CHAPTER VI

AN EVALUATION OF DBLY VISUALIZATOIN ON REAL DATASET

In Chapter V, we described pilot study based on synthetic datasets. In this chapter, we present the details of the main study, which is based on a real hyperspectral dataset, i.e., AVIRIS Lunar Lake image.

6.1 Experimental Design and Settings

6.1.1 Datasets and Colors

Since it is difficult to describe and quantitatively measure patterns in real datasets, we embedded features to facilitate the edge detection task in a user study. We rotated and flipped the image to inhibit learning by participants due to having seen the images with other visualization techniques.

Embedding Features:

Since it is difficult to get the ground truth for the perceptual edges in the Lunar Lake dataset, we embedded a gradient bar into the classified results as shown in Figure 6.1. The image size is 200 × 200; the start position of the bar varies from 10 to 40 pixels from left to right.

Rotating and Flipping Images:

To inhibit learning the image pattern by the participants, seven more datasets were generated by flipping and rotating the Lunar Lake dataset. Figure 6.2(a) is the sample image of SOFT visualization of original dataset, and Figure 6.2(b) is the *x*-axis flipped HARD visualization.

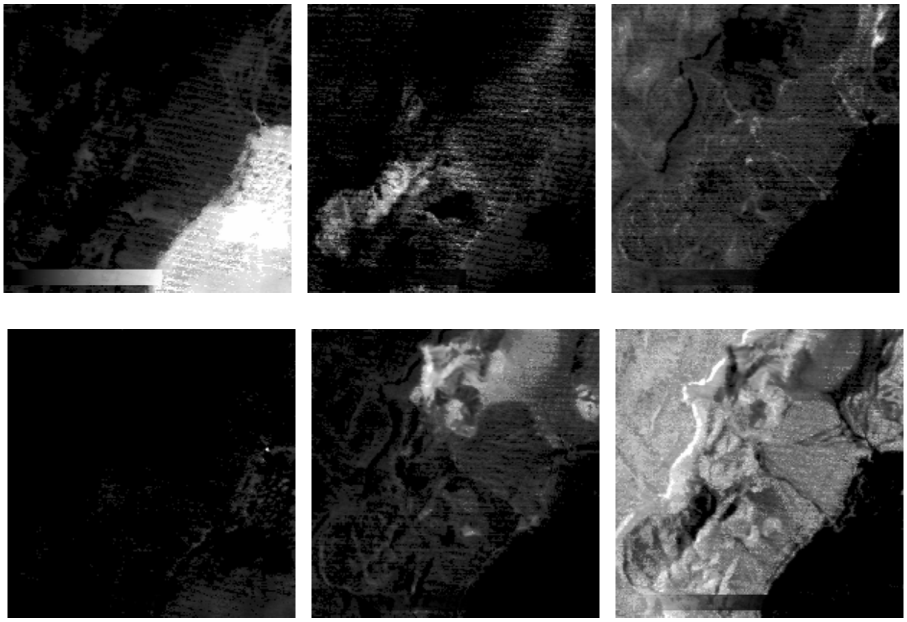
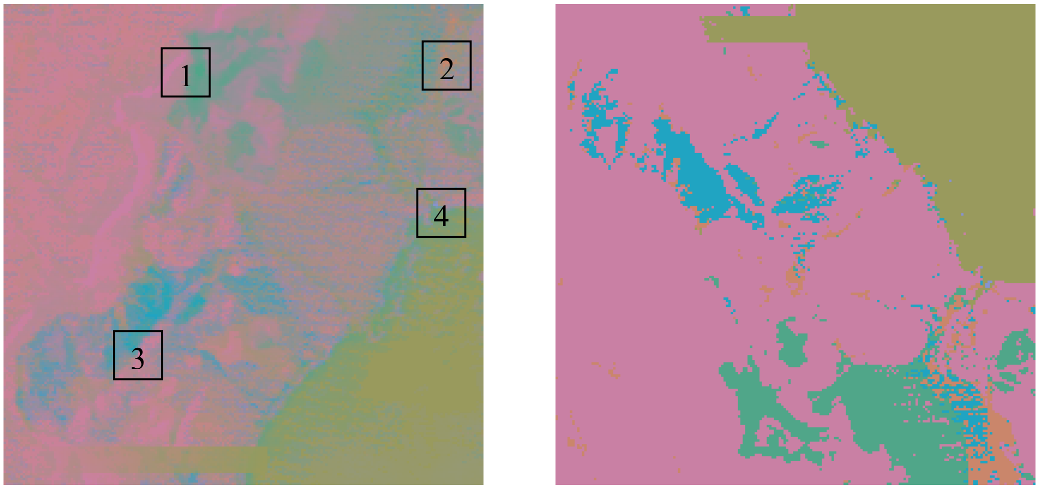


Figure 6.1 The Lunar Lake abundance images with embedded features.



(a) (b)

Figure 6.2 Sample images used in the main user study. (a) Original dataset, SOFT visualization; (b) *x*-axis flipped, HARD visualization.

Four 20 × 20 blocks, which are indicated by the black boxes in Figure 6.2(a), were selected to simulate the zooming-in operation in visualization. These four blocks were chosen because they are not dominated by one or two classes. Figure 6.3 shows the sample images of these four blocks. From left to right, and top to bottom, the four images correspond to the four blocks in Figure 6.2(a), and represent the four visualization techniques, GRAY, HARD, SOFT, and DBLY, respectively.

The same colors as in the pilot study were used in this study (Figure 5.1). In order to maximize the discrimination of the neighboring classes, the colors are assigned to neighboring images in a way that perceptual distance is the largest.

6.1.2 GUI and Software

A similar GUI has been implemented to perform this study. As shown in Figure 6.4(a), the view space was split into a 3 × 2 matrix to display the six classes of Lunar Lake. The image was displayed at the center of view space as Figure 6.4(b) when the test images are in color (HARD, SOFT, and DBLY). For each participant, a list of predetermined tests was displayed by the GUI. Answers and time spent in answering each question were recorded by the GUI and written to an output file for further analysis.

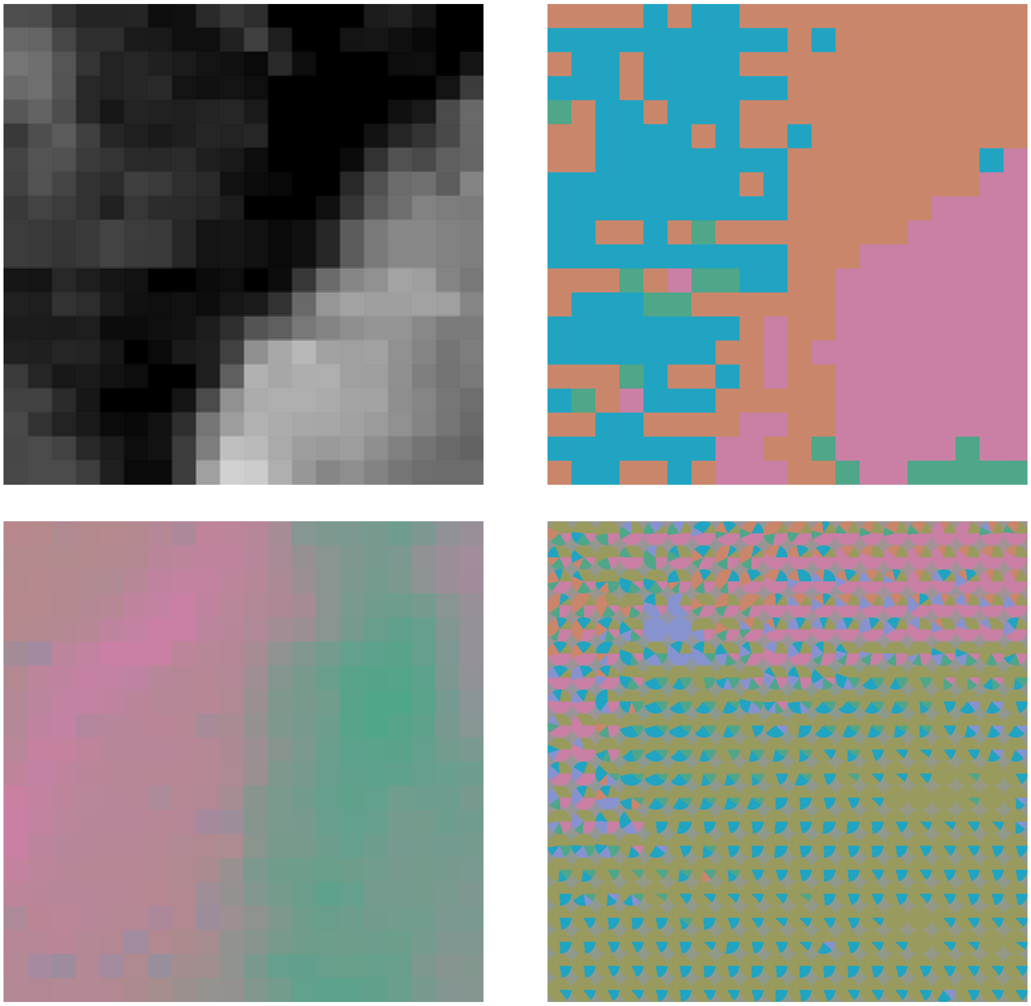
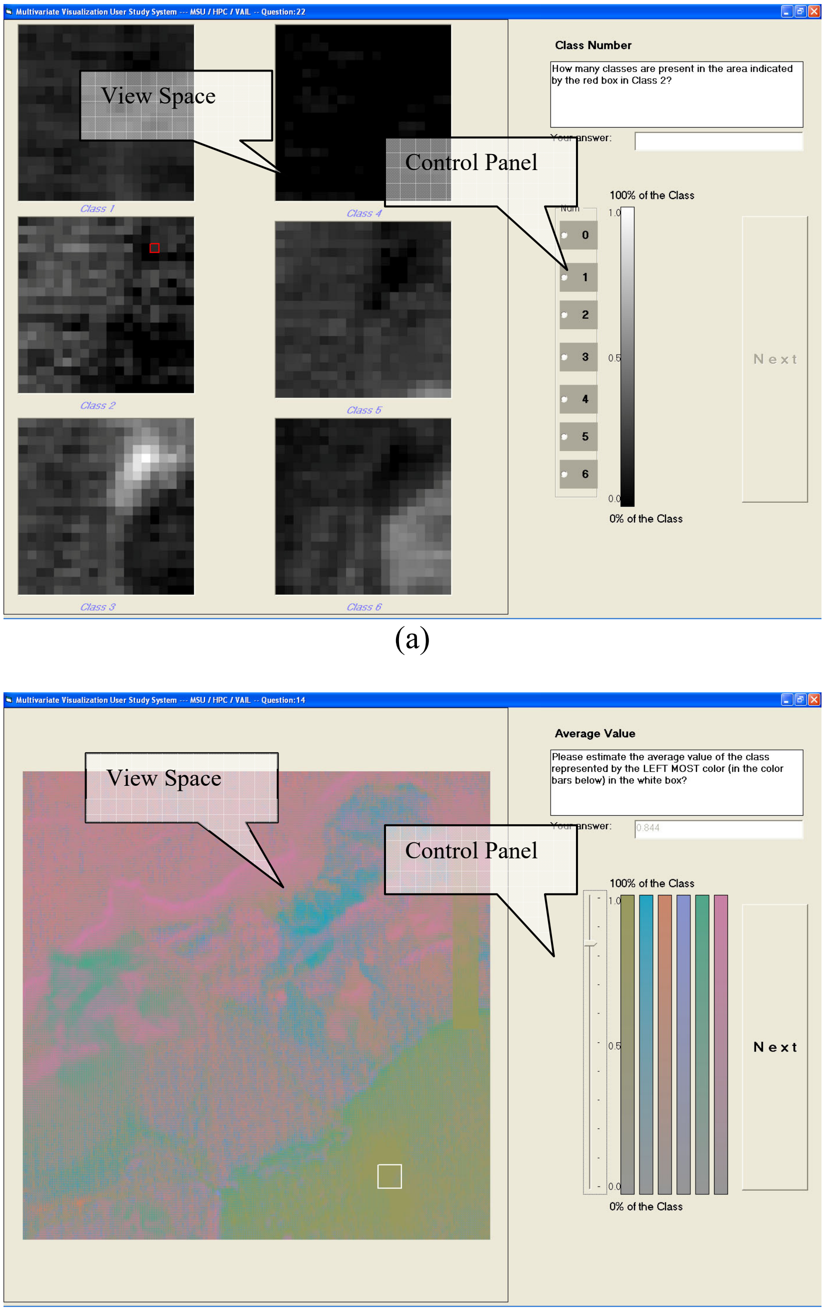


Figure 6.3 Sample images, which are indicated by the black box in Figure 6.2(a), are used to simulate the zooming-in operation in the main user study. From left to right and from top to bottom, the four images correspond to the four blocks in Figure 6.2(a), and represent the four visualization techniques, GRAY, HARD, SOFT, and DBLY, respectively.



Multivariate Visualization User Study System --- MSU / HPC / VAIL -- Question:22

Class Number

How many classes are present in the area indicated by the red box in Class 2?

Your answer:

(a)

Multivariate Visualization User Study System --- MSU / HPC / VAIL -- Question:14

Average Value

Please estimate the average value of the class represented by the LEFT MOST color (in the color bars below) in the white box?

Your answer: 0.844

(b)

Figure 6.4 Samples of the GUI in training mode in the main study. (a) The GUI with six gray-scale images to estimate the class number (class recognition task) at the pixel indicated by the red box in class 2; (b) the GUI with one SOFT image to estimate the average value of the class represented by the left most color bar in the white box.

6.1.3 Participants

A total of fifteen participants (i.e., four research associates, one professor, and ten graduate students) participated in this user study. The four research associates are remote sensing experts and two of the graduate students' major research area is remote sensing. They are categorized as remote sensing specialists. One participant is a professor from Department of Computer Science. One graduate student is in visualization. The rest are in chemistry (3), agriculture economics (2), aerospace engineering (1), and industrial engineering (1) whose research is related to human factors.

We administered the test with the same laptop computer used in the pilot study, with its display profile being set to the standard RGB color space. The study was run in a conference room with only the experimenters and the participant present. Before the study, participants completed necessary paperwork and answered a general questionnaire. We paid each participant $10 for completing the study. To reduce any potential training bias, we wrote a training guide so that all participants received the same training. Participants were not required to finish all the training tests; we allowed them to start the experiment early if they desired. We encouraged participants to ask questions during the training. Testing continued for 30~40 minutes. To avoid fatigue effects, participants could take a break after every 30 tests. Participants were not required to finish all the tests. The entire study took about an hour, including the paperwork and training.

Table 6.1

THE INDEPENDET VARIABLES AND DEPENDETN VARIABLES STUDIED IN THE MAIN STUDY.

<table>
<tr><th colspan="3">INDEPENDENT VARIABLES</th></tr>
<tr><td><i>participant</i></td><td>15</td><td>(random variable)</td></tr>
<tr><td><i>technique</i></td><td>4</td><td>GRAY,HARD, SOFT, DBLY</td></tr>
<tr><td><i>task</i> ×
<i>subtask</i> ×
<i>num datasets</i> ×
<i>repetition</i></td><td>152</td><td>perceptual edge location × 1 × 8 × 3
block value estimation × 5 × 8 × 1
Class recognition × 6 × 4 × 2
target value estimation × 5 × 4 × 2</td></tr>
<tr><th colspan="3">DEPENDENT VARIABLES</th></tr>
<tr><td><i>response time</i></td><td colspan="2">measured for each test in seconds</td></tr>
<tr><td><i>relative error</i></td><td colspan="2"><i>user answer – ground truth</i></td></tr>
<tr><td><i>absolute error</i></td><td colspan="2">| <i>relative error</i> |</td></tr>
<tr><td><i>normalized error</i></td><td colspan="2">$\frac{user\ answer - ground\ truth}{bar\ length} \times 100\%.$
(measured for perceptual edge detection task)</td></tr>
</table>

6.1.4 *Independent and dependent variables*

Similar dependent and independent variables as in the pilot study were measured as listed in Table 6.1. The independent variables included *participant, technique,* and the *tuple (task* × *subtask* × *dataset* × *repetition)*. Each task was built upon either 4 or 8 datasets; there were a total of 152 tuples. The experiment consisted of 4 techniques × 152 tuples = 608 possible tests.

The quantified dependent variables were *response time, relative error, absolute error,* and *normalized error*. Standard error plots, ANOVA, post-hoc analysis, and *z-score* were employed to analyze dependent variables. The detailed analysis is discussed in the following section.

6.2 Experimental Results and Discussions

In this section, we present the details of the four tasks and discuss the results from each task. We determine the outliers and analyze the results as in the pilot study.

6.2.1 Global Pattern Display Capability

After conducting the synthetic data study, we realized it is obviously easier to detect relative location if objects are in one image, so we did not repeat the relative position task using real data. We only conducted the perceptual edge detection and block value estimation task on the full image data.

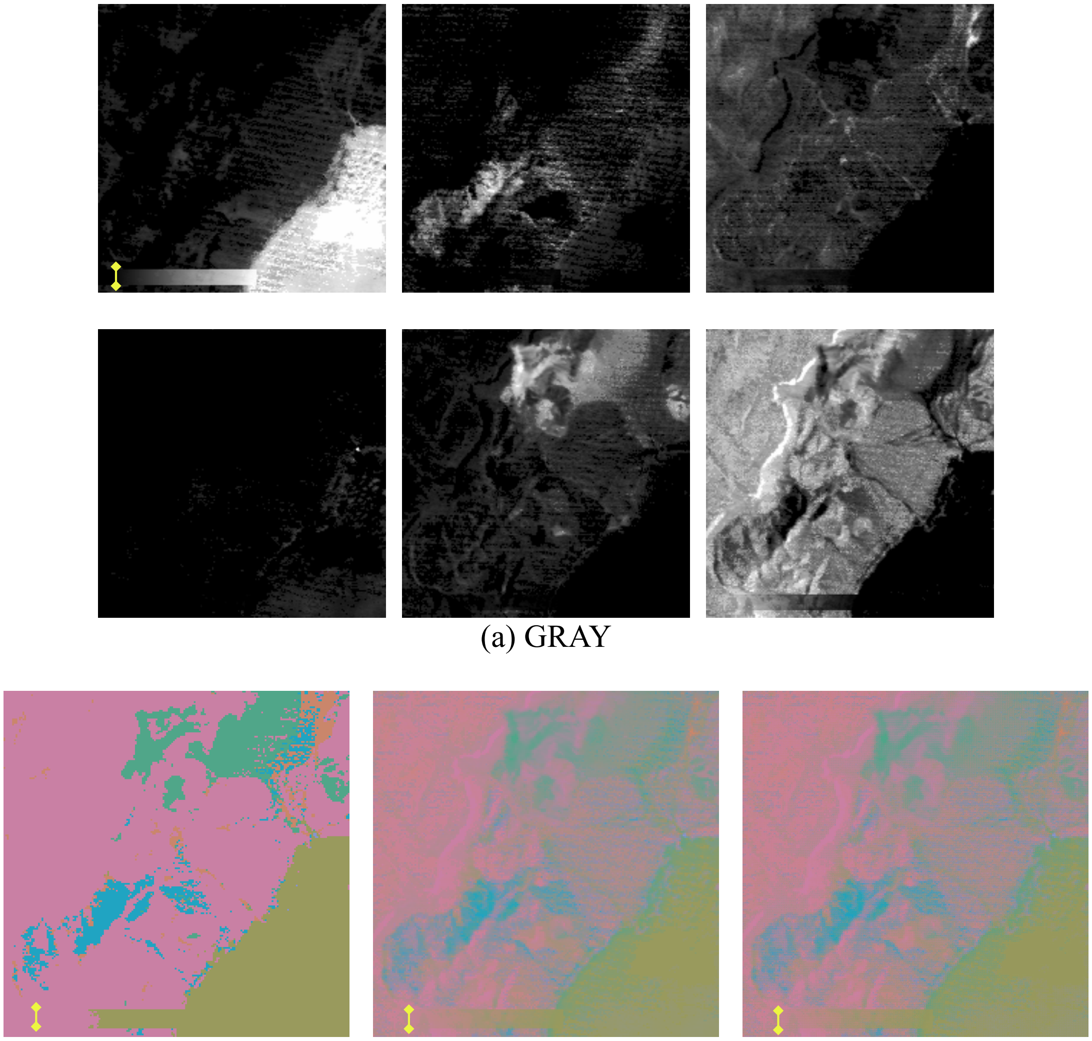
Perceptual Edge Detection

This task was designed to test how well a participant can detect material edges using each visualization technique.

Task: The sample images are shown as Figure 6.5. For all the images, the matching pixels still satisfy the non-negativity and sum-to-one constraints. The sample questions of this task are:

GRAY: *At what position along the line segment does the quantity of CLASS i go to zero as you move down/up/left/right from the point O? Please click at the position.*

Others: *At what position along the line segment does the quantity of the material represented by the LEFT MOST color (in the color bars below) go to zero as you move down/up/left/right from the point O? Please click at the position.*



(a) GRAY

(b) HARD

(c) SOFT

(d) DBLY

Figure 6.5 An example of the perceptual edge detection task. The yellow lines indicate the ground truth positions.

Table 6.2

OUTLIERS FOR PERCEPTUAL EDGE DETECTION TASK.

	GRAY	HARD	SOFT	DBLY	Total
answers	104	101	131	116	452
normalized error	9	7	12	6	34
response time	9	4	8	6	27

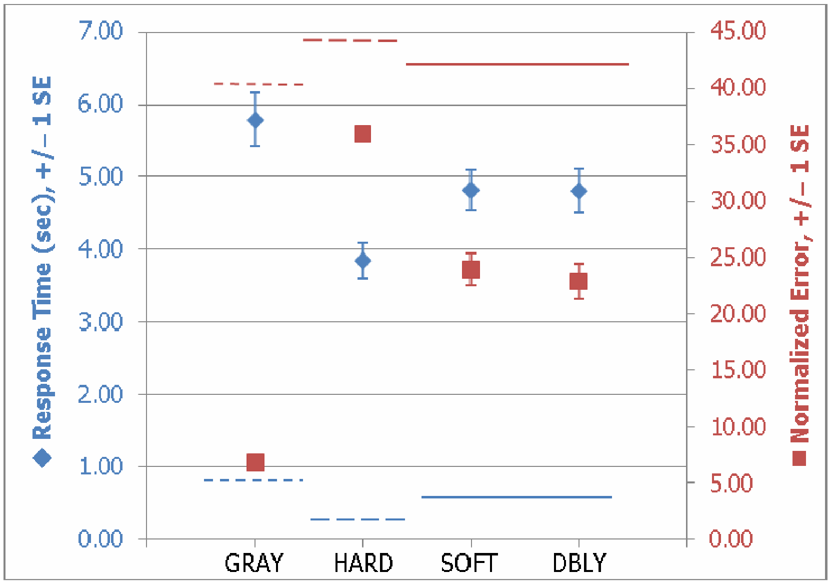


Figure 6.6 Result of the perceptual edge detection task. The lines indicate the results of post-hoc comparisons.

Results: We recorded the coordinate of the user's mouse click for this task and measured the dependent variables: *response time* in seconds and *normalized error*. A total of 452 answers were recorded, including 104 answers for GRAY, 101 for HARD, 131 for SOFT, and 116 for DBLY. The outliers were listed in the Table 6.2. The means displayed in Figure 6.6 indicate that participants have the best performance to recognize the perceptual edges for GRAY. SOFT and DBLY are in the second rank. HARD is once again at the highest error level as it was in the pilot study, but participants took the shortest time to figure out the answer. *F*-value and *p*-value analysis found a major effect of visualization technique on *normalized error* ($F(3, 42) = 166.49$, $p < .000$) and *response time* ($F(3, 42) = 12.94$, $p < .000$). The post-hoc comparisons indicated the differences as shown in Figure 6.6. The high *normalized error* of HARD comes from the hard classification itself. Since there are six classes in the study, value less than 1/p (p is the number of the classes) will not be the largest.

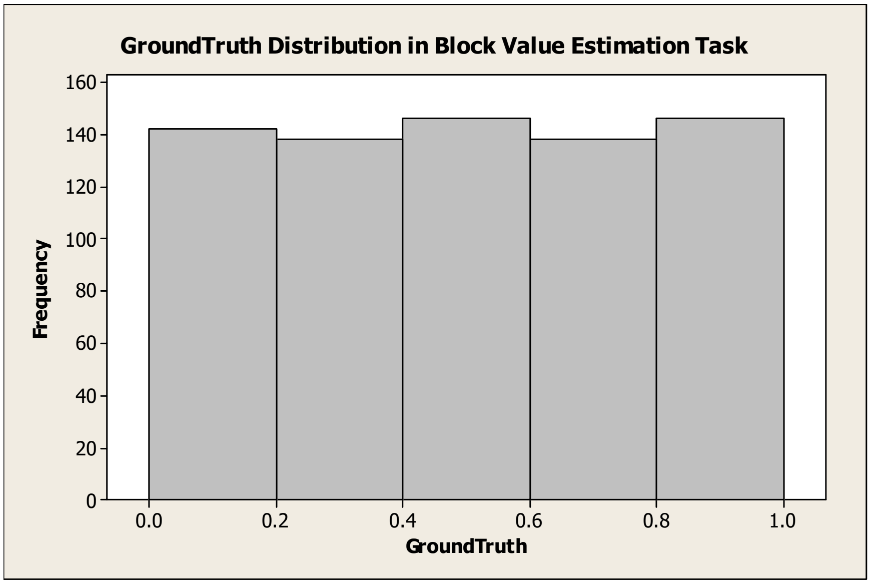


Figure 6.7 The distribution of ground truth for block value estimation task. The labels below the ground truth axis indicate the five ground truth subsets.

Block Value Estimation

Color mapping plus legends is widely used in visualization. This task was designed to assess participants' ability to accurately read the continuous values encoded by a color. In each region, colors represent overlapped multiple scalars.

Task: This task was generated by the following steps:

1) Dividing the range of data value [0.0, 1.0] into five subsets: s ={ [0.0, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6,0.8], [0.8, 1.0]};

2) Initialize the subset s_i as the first subset s_1 = [0.0, 0.2];

3) Randomly generate a 20 × 20 pixel block $\boldsymbol{B}$ and Class index k;

4) If the average value v of the k-th class in $\boldsymbol{B}$ is in s_i ($v \in s_i$), then write the class index k and block $\boldsymbol{B}$ as an input task; otherwise, check the $(k+1)$-th class;

5) If a task has written to the input file, assign $i = i+1$ and then go back to step 2; otherwise, repeat step 3 until a task is written to the input file.

6) Permute the task sequence in the input file.

Figure 6.7 shows the distribution of the ground truth of the tasks which have been answered by the participants. The plot shows that the ground truth is well balanced for the five subsets.

The task was to match the block color with a slider that contains the color of each of the six endmembers. A sample dataset is displayed in Figure 6.8. In the sample images, the participants were asked to estimate the average value in the red block in the sixth class (right-bottom in Figure 6.8 (a)). The sample questions of the task are:

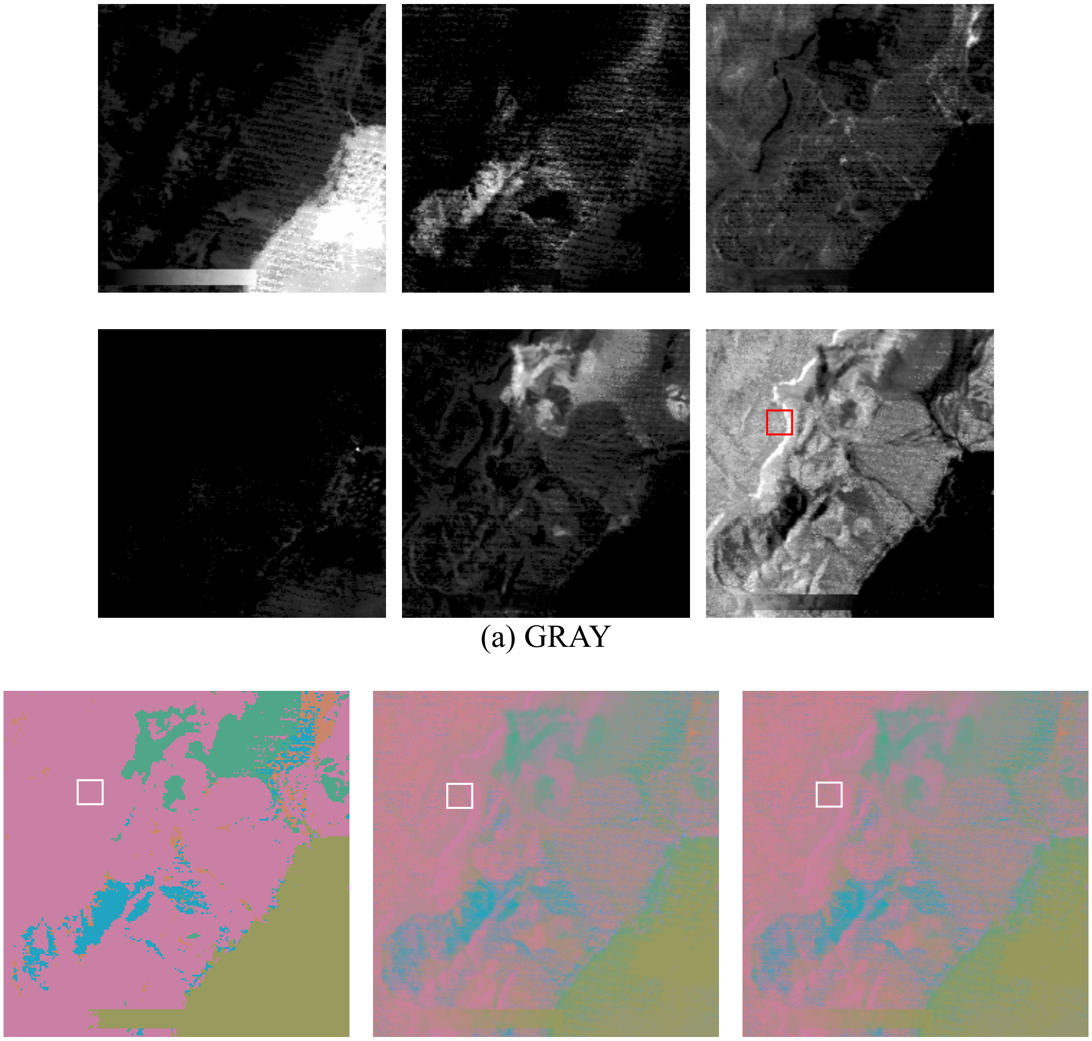
GRAY: Please estimate the average value of CLASS i in the red box in CLASS i.

Others: Please estimate the average value of the class represented by the LEFT MOST color (in the color bars below) in the white box?

Table 6.3

OUTLIERS FOR BLOCK VALUE ESTIMATION TASK.

	GRAY	HARD	SOFT	DBLY	Total
answers	178	183	167	182	710
absolute error	11	14	7	20	52
response time	5	18	4	10	37



(a) GRAY

(b) HARD

(c) SOFT

(d) DBLY

Figure 6.8 A sample of the block value estimation task. The red/white box indicates the position of target blocks. Figure 6.4(b) shows the GUI for this task.

Results: We recorded a total of 710 answers, including 178 answers for GRAY, 183 for HARD, 167 for SOFT, and 182 for DBLY. A total of 52 *absolute error* outliers and 37 *response time* outliers were replaced by the median of remaining answers. The detailed information of outliers is listed in Table 6.3. After the outliers have been removed, the maximum *absolute error* is 58.02%.

The *F*-value and *p*-value tests found a major effect of visualization technique on *absolute error* (*F*(3, 42) = 16.46, *p* < .000), and *response time* (*F*(3, 42) = 19.81, *p* < .000). The post-hoc comparisons indicated the differences as shown in Figure 6.9; the HARD technique was significantly worse than other techniques.

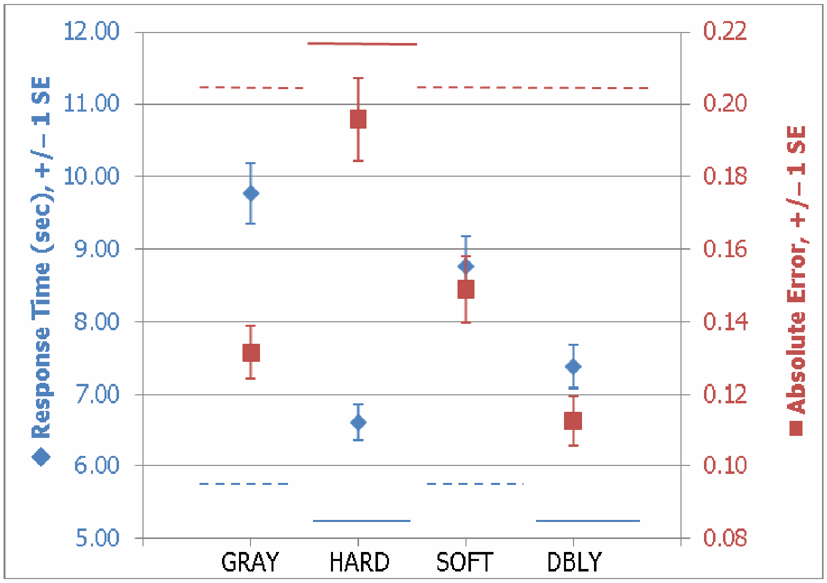
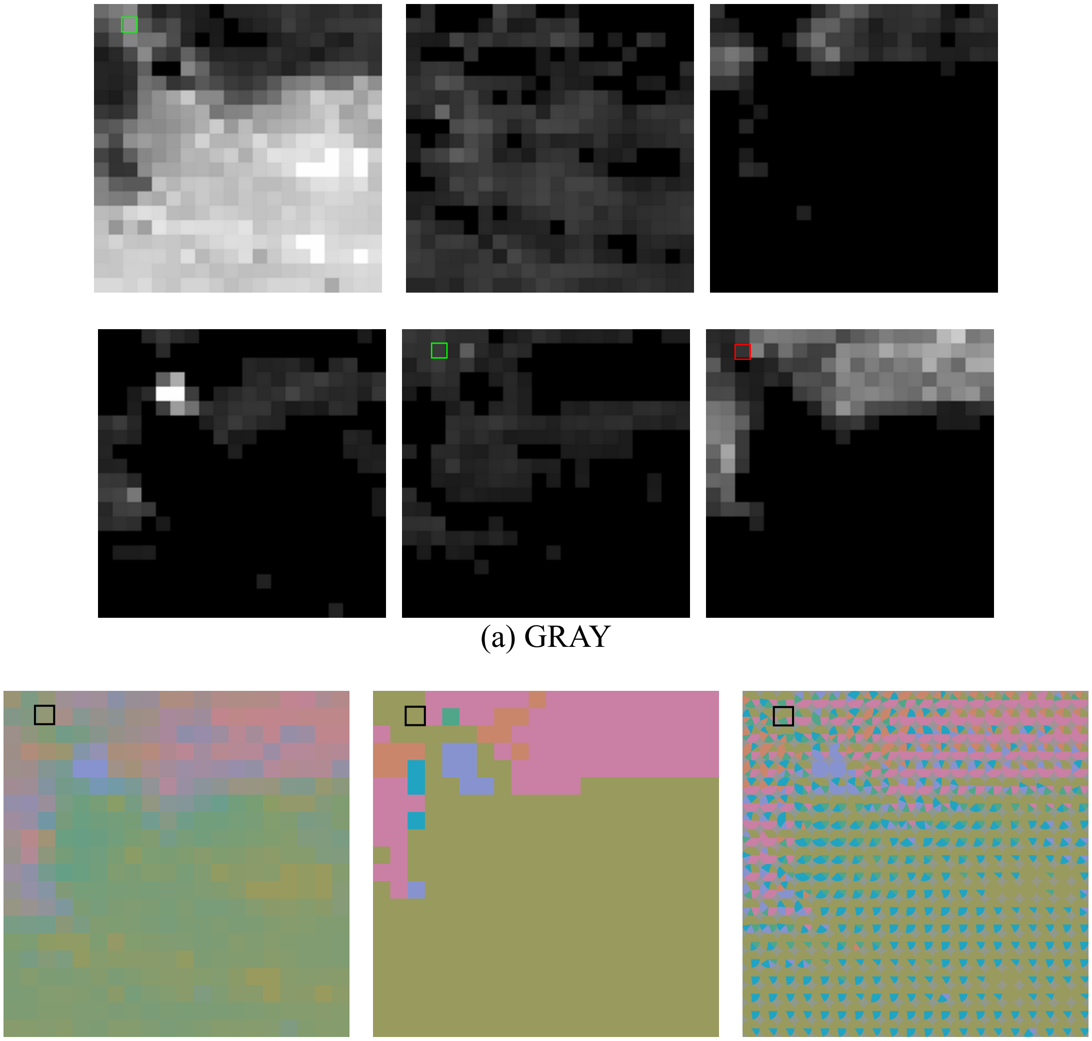


Figure 6.9 Result of the block value estimation task. The lines indicate the results of post-hoc comparisons.

6.2.2 *Local Information Conveying Capability*

In this section, we designed two tasks, class recognition and target value estimation, to evaluate the capability of conveying detailed information at the subpixel level. Four blocks of size 20 × 20 pixels (shown in Figure 6.1(a)) were cut to simulate the zooming-in operation, and blending parameter was set to 0.8 in DBLY to pop out the pie-chart layer.



(a) GRAY

(b) HARD

(c) SOFT

(d) DBLY

Figure 6.10 A sample set of images for testing class recognition with 3 classes in the pixel under test. For the GRAY visualization, the testing pixel position is indicated by the red box in the right-bottom gray image. The co-existing classes are marked by the green box. During the real test, the green boxes are not displayed. Black box marks the testing position in other visualization techniques.

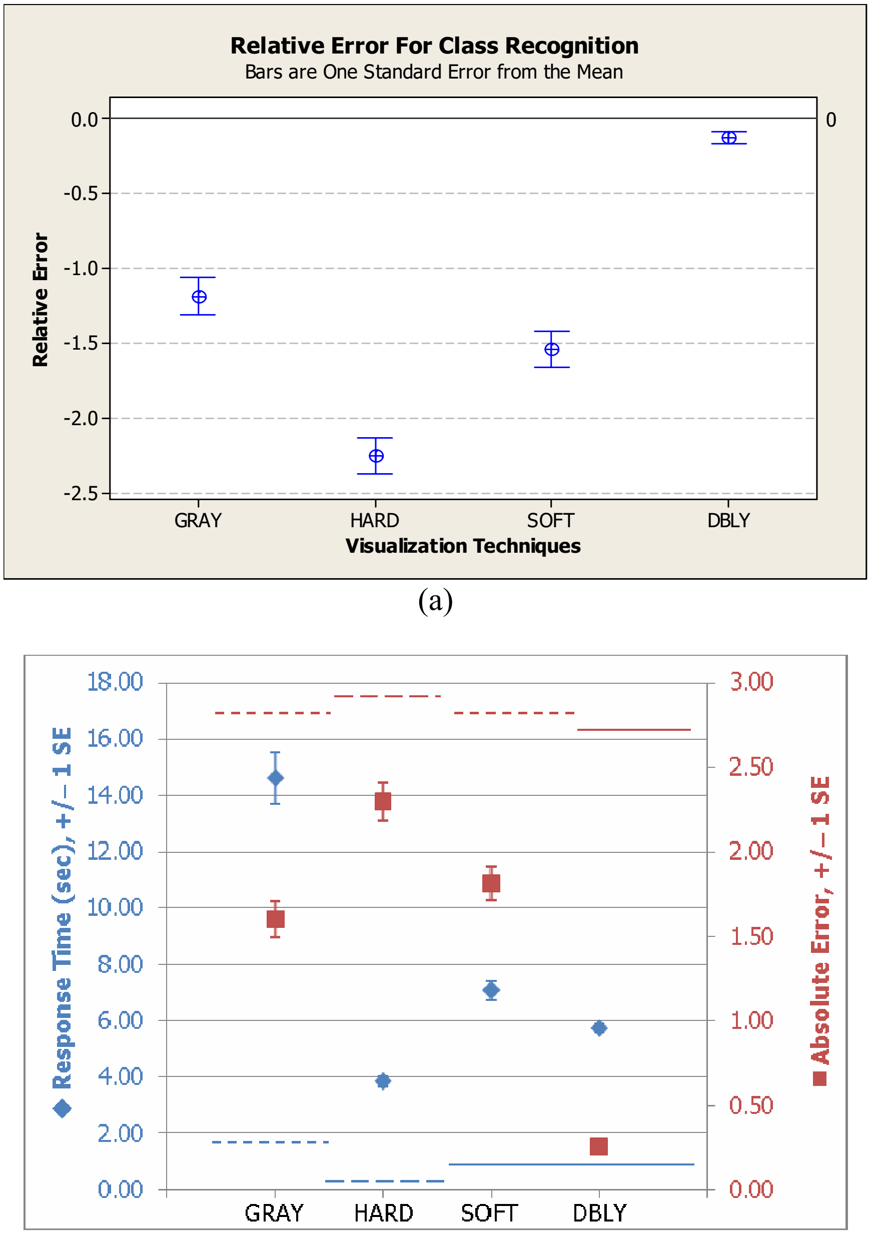
Class Recognition

Recognizing the number of classes is the first step in understanding the dataset. This goal of this task was to assess participants' ability of determining the number of endmembers present when zooming into an image.

Task: Four datasets with size of 20 × 20 pixels were cut to simulate the zooming-in operation. Each pixel may contain one or more classes. Each pixel is a square with GRAY, HARD, and SOFT, and a square covered by a pie-chart in DBLY. In this task, we asked participants to indicate the number of classes present in the given pixel. Figure 6.10 displays a sample dataset. In the GRAY visualization (Figure 6.10 (a)), the white pixel contains 100% of that class; otherwise, other classes co-exist in that pixel. In the HARD visualization, the color of the pixel is represented by the color of the class whose value is the maximum in the pixel. In the SOFT visualization, the color of the pixel is the combined color of all the classes existing in the pixel. In DBLY visualization, the different colors in the fan-shape region represent different classes and an angle of fan-shape represents the percentage of the corresponding class in the pixel. The sample questions for this task are:

GRAY: How many classes are present in the area indicated by the red box in Class i?

Others: How many classes are present in the white box?



(a)

(b)

Figure 6.11 Result of the class recognition task. (a) *Relative error* with error bars; (b) *response time* and *absolute error* with error bars. The color-coded lines indicate the result of post-hoc comparisons.

Results: The dependent variables of *response time, relative error*, and *absolute error* were measured for this task. The *relative error* ranged from −5 to 5. The negative *relative error* means the participants underestimate the class number; the positive *relative error* means that participants overestimate the class number; and zero means no error. Therefore, the range of *absolute error* is [0, 5]. We recorded a total of 834 answers, including 209 for SOFT, 207 for DBLY, 208 answers for GRAY and 209 for HARD. There were 5 *relative error* outliers in GRAY and a total of 32 *response time* outliers. All outliers were replaced by the median of remaining answers. Detailed information of the outliers is listed in Table 6.4.

We found that participants trend to underestimate the number of classes in all the tested visualization techniques (Figure 6.11(a)). Because one color is displayed in the HARD visualization, it is basically impossible for participants to figure out the number of the co-existing classes. Hence HARD brings about the largest error in both *relative error* and *absolute error* (Figure 6.11(b)). Several participants indicated that they resorted to guessing, which explains the relatively low *response time* for this technique.

Table 6.4

OUTLIERS FOR CLASS RECOGNITION TASK

	GRAY	HARD	SOFT	DBLY	Total
answers	209	209	209	207	825
relative errors	5	0	0	0	5
response time	4	9	9	10	32

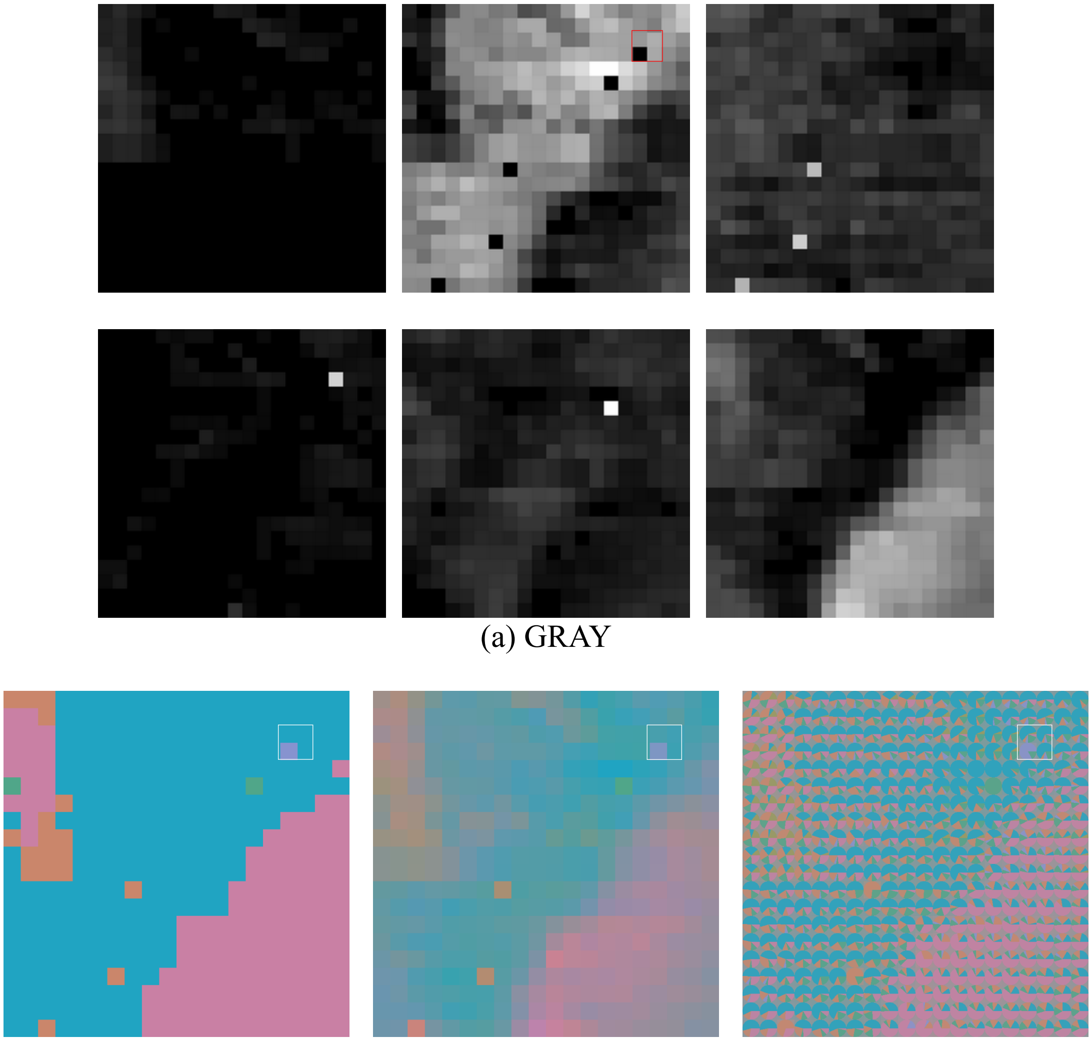
GRAY and SOFT provide some clues for participants to speculate the ground truth. Even these clues do not provide the precise information; participants can determine the answer by estimating the pixel locations in other abundance images in GRAY and by considering the appearance of the mixed color in SOFT. These explain why GRAY and SOFT achieve better performance than HARD in both *relative error* and *absolute error*. However, participants took longer time to finish the class estimation from the GRAY images. When conducting the study, we found that some participants tried to accurately relocate the pixels in each image in order to align the side-by-side displayed images. This indicates that it is better to provide an automatic tool to align the pixels for side-by-side visualization.

We found a major effect of visualization technique on *relative error* ($F(3, 42) = 73.49, p<.000$), *absolute error* ($F(3, 42) = 91.65, p < .000$), and *response time* ($F(3, 42) = 101.93, p < .000$). As shown in Figure 6.11(b), the post-hoc comparisons indicated the differences for both *response time* and *absolute error*.

Since different colors in the pie-charts represent different classes, participants can get very precise information at subpixel level. The results show that DBLY can achieve smaller *relative error* (−0.2) and *absolute error* (0.2) compared to other techniques and the response from participants was significantly faster.

Target Value Estimation

Task: This task was designed to evaluate the ability of the four techniques to convey quantitative information. It is very similar to the block value estimation task. The size of the datasets is 20 × 20 and the target block is a 2 × 2 pixel block. The average value in the target block varies in the range s_i, where s_i belongs to the set {[0.0, 0.2], [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], [0.8, 1.0]}. The process of generating the task tuple is the same as in



(a) GRAY

(b) HARD (c) SOFT (d) DBLY

Figure 6.12 An example of the target value estimation task. The red/white boxes indicate the target position.

the block value estimation task. The task was to match the block color with a slider that contains the color of each of the 6 classes. Figure 6.12 displays an example dataset, where the average value of Class 2 in the target block is in the range [0.2, 0.4]. The distribution of the ground truth indicates that the ground truth is equally distributed over the range of [0.0, 1.0] with interval size of 0.2 (Figure 6.13). The sample questions for this task were:

GRAY: Please estimate the average value of CLASS i in the red box in CLASS i.

Others: Please estimate the average value of the class represented by the LEFT MOST color (in the color bars below) in the white box?

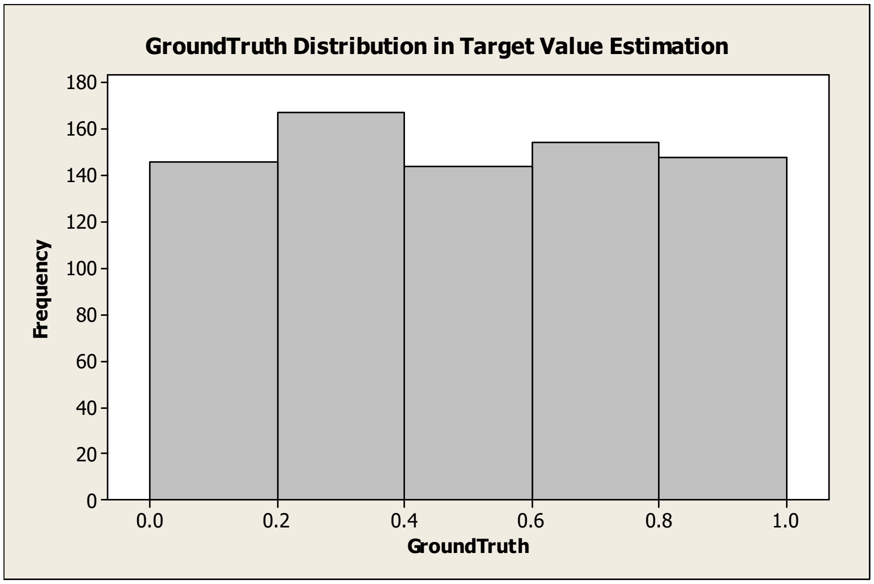


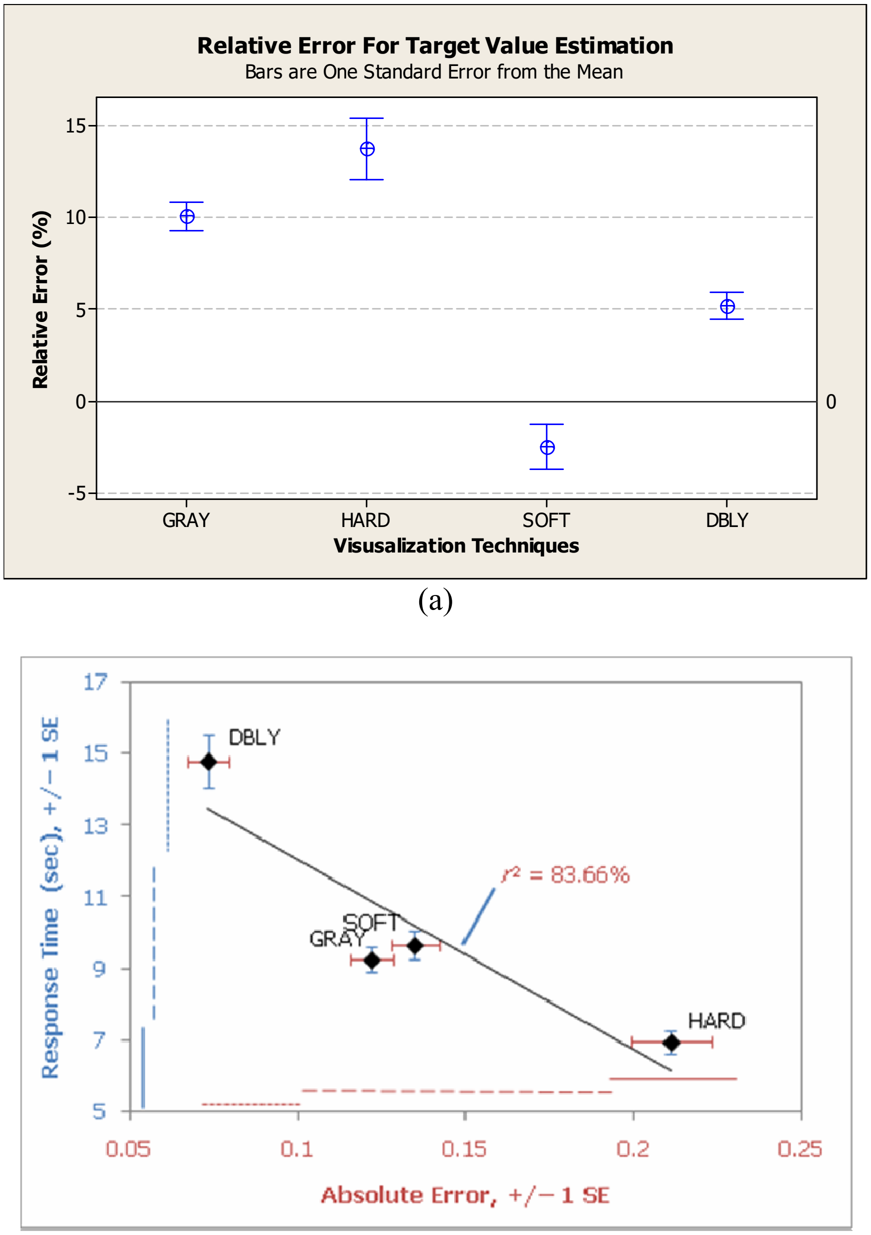
Figure 6.13 The distribution of ground truth for target value estimation task.

Table 6.5

OUTLIERS FOR TARGET VALUE ESTIMATION TASK

	GRAY	HARD	SOFT	DBLY	Total
answers	187	195	188	189	759
relative error	13	1	17	12	43
response time	2	2	8	2	14

Results: The dependent variables calculated for this task were *response time*, *relative error*, and *absolute error*. The *relative error* ranged from -34.8% to 75%. The negative *relative error* means participants underestimate the value; the positive *relative error* means participants overestimate the value; and zero means no errors. The *absolute error* ranged from 0 (no error) to 75% (maximum error).



(a)

(b)

Figure 6.14 Result of estimating the target value estimation task. (a) Mean and interval plot for *relative error*; (b) *absolute error* and *response time* in seconds. The color-coded lines indicate the result of post-hoc comparisons.

We recorded a total of 759 answers, including 187 answers for GRAY, 195 for HARD, 188 for SOFT, and 189 for DBLY. A total of 43 *relative error* outliers and 14 *response time* outliers were corrected. The outliers' number is listed in Table 6.5.

The *relative error* plot (Figure 6.14(a)) indicates the participant trends to overestimate the target value in GRAY, HARD, and DBLY visualization techniques, but underestimate the value in SOFT visualization. The color of the pixel in HARD is chosen by the maximum likelihood theory. A class with value greater than $1/p$ (p is the number of classes) in a pixel has high potential to be chosen as the dominating class and its color will be assigned to the pixel. This results in the participants overestimating the value in the HARD visualization techniques. In GRAY visualization, the maximum value (1.0) is mapped to white and the minimum value (0.0) is mapped to black. The dark background increases the contrast of gray-scale images and makes participants overestimate the target value in GRAY visualization. This study did not provide the reason that participants overestimate the value in DBLY visualization. One possible reason may be that the size of the pie-chart is not big enough for participants to read detailed information.

Comparing to the overestimation in the block value estimation task, participants trend to underestimate the value of SOFT. In the global display, the high value surrounding the target block may affect participants' judgment. In the local zooming-in operation, participants lack the global view and trend to underestimate the value, as they did in the class recognition task.

We found a major effect of visualization technique on *relative error* ($F(3, 42) = 38.15, p<.000$), *absolute error* ($F(3, 42) = 48.09, p < .000$), and *response time* ($F(3, 42) =$

62.11, p < .000). The post-hoc comparisons indicated the differences shown in Figure 6.14 for both *absolute error* and *response time*.

Figure 6.14(b) shows a clear *response time*/accuracy (*absolute error*) tradeoff for GRAY, SOFT, HARD, and DBLY. Participants' responses were very accurate with DBLY, but it took them longer to read the individual pie charts. GRAY and SOFT required consideration about color combination. The task was almost impossible with HARD, so participants provided the straightforward answer based on the average pixel color values.

6.3 Conclusion

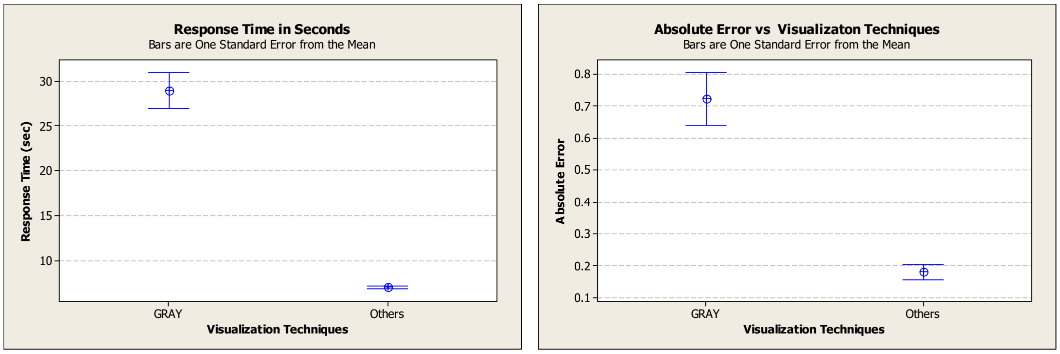
The evaluation on the real dataset confirms that the GRAY approach is not sufficient to locate the relative location (high absolute error rate in class recognition task) or to visualize the local detailed information, the GRAY method is effective to display the perceptual edge and for participants to estimate the block value. As concluded in the pilot study, this main study also confirms that the HARD approach is less effective for perceptual edge detection, block value estimation, or local information display. The SOFT approach is the best for block value estimation as in the pilot study, but it is less efficient for displaying local information.

Both the pilot study and the main study demonstrate that the DBLY technique is the most accurate method for showing local details. Taking the advantages of the SOFT method, the DBLY algorithm is also effective in displaying global patterns. Both studies validate that adding a pie-chart layer to the SOFT approach is necessary for conveying local information while the DBLY algorithm maintains efficiency in displaying global patterns that was exhibited by the SOFT method.

CHAPTER VII

ANALYSIS AND DISCUSSION

In the previous two chapters, we described the details of two user studies and the results. In this chapter, we analyze and discuss those studies and the results. Later, we present the advantages and disadvantages of the DBLY algorithm.



(a) (b)

Figure 7.1 The result of displaying objects in one image vs. displaying objects in separate images. “GRAY” represents the techniques which display objects in separate images and “Others” in one image.

7.1 Discussions on Studies

7.1.1 Hypotheses

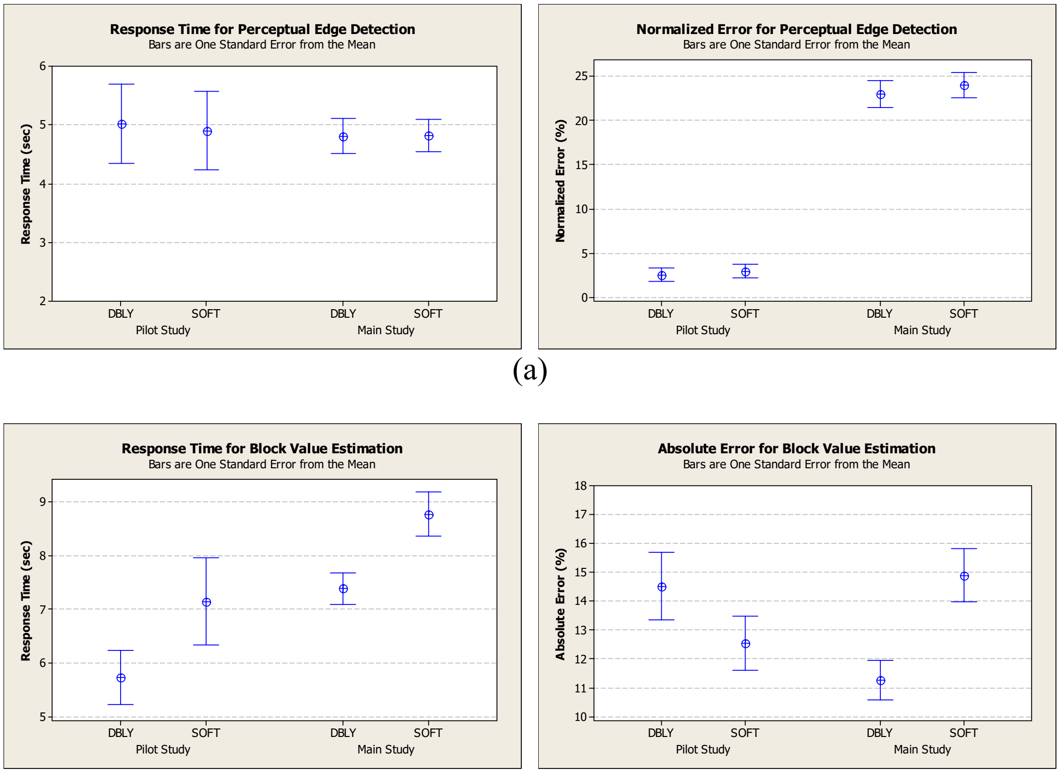
In Chapter V, we introduced three hypotheses which our user studies were to test. The task of relative location in the pilot user study was designed under **H1**; both perceptual edge detection and block value estimation tasks were designed under **H2**; and class recognition task and target value estimation tasks were designed under **H3**. In this section, we fully discuss these hypotheses.

H1: True. Displaying materials in one image is better than displaying them in several side-by-side images when trying to ascertain relative location of materials.

Figure 5.4 indicated that the visualization techniques significantly impact the results. We combined the results of the HARD, SOFT, and DBLY techniques, which display objects in one image, into one group: Others. The comparison results are shown in Figure 7.1. Both *response time* and *absolute error* measurements indicate that displaying materials in one image has better performance than side-by-side when trying to ascertain relative location of two materials. The *F*-value and *p*-value analysis demonstrated that different visualization techniques have significantly different performance in determining the relative location of two objects (*response time*: $F(1, 9) = 42.27$, $p<.000$, and *absolute error*: $F(1, 9) = 14.73$, $p<.004$).

H2: True. Adding a pie-chart layer does not degrade the background layer (SOFT)'s capability in displaying global patterns.

Figure 7.2 shows the comparison of perceptual edge detection and block value estimation tasks. It indicates that there is no significant difference between SOFT and DBLY visualization techniques for these tasks. The average time for participants to conduct perceptual edge detection is around 5 seconds. However the *normalized error* increased approximately 20% from the pilot study to the main study. This difference indicates that the number of classes or datasets may have significant impact on the performance in determining perceptual edges. Figure 7.2(b) indicates that participants' performance in estimating block value using DBLY is not worse than using SOFT. The average *response time* for DBLY and SOFT is 6.88 seconds and 8.24 seconds, respectively. The average mean of the *absolute error* is 12.25% for DBLY and 14.13% for SOFT. Therefore, adding a pie-chart layer to the SOFT visualization techniques to form the DBLY does not influence the display of global patterns.



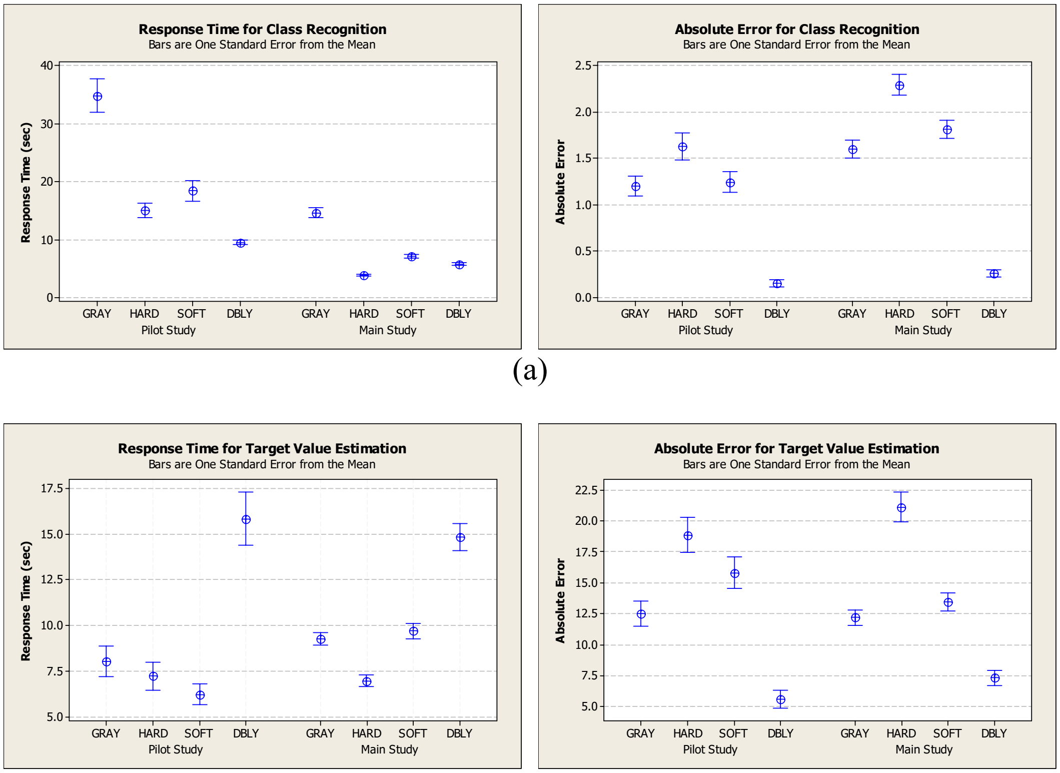
(a)

(b)

Figure 7.2 The comparison of SOFT and DBLY visualization. (a) *Response time* and *normalized error* for perceptual edge detection task; (b) *response time* and *absolute error* for block value estimation task.

H3: True. The Pie-chart layer increases DBLY's efficiency in revealing local information.

Figure 7.3 shows the results of comparison of the two user studies on local information display. Both studies indicated that the DBLY is significant better than other techniques on precisely visualizing local information. Compared to the SOFT visualization techniques, the pie-chart layer meaningfully increases the capability of DBLY in more precisely visualizing local information. However, there is a tradeoff between the *response time* and accuracy on DBLY due to the mental calculations required for individual pie-charts.



(a)

(b)

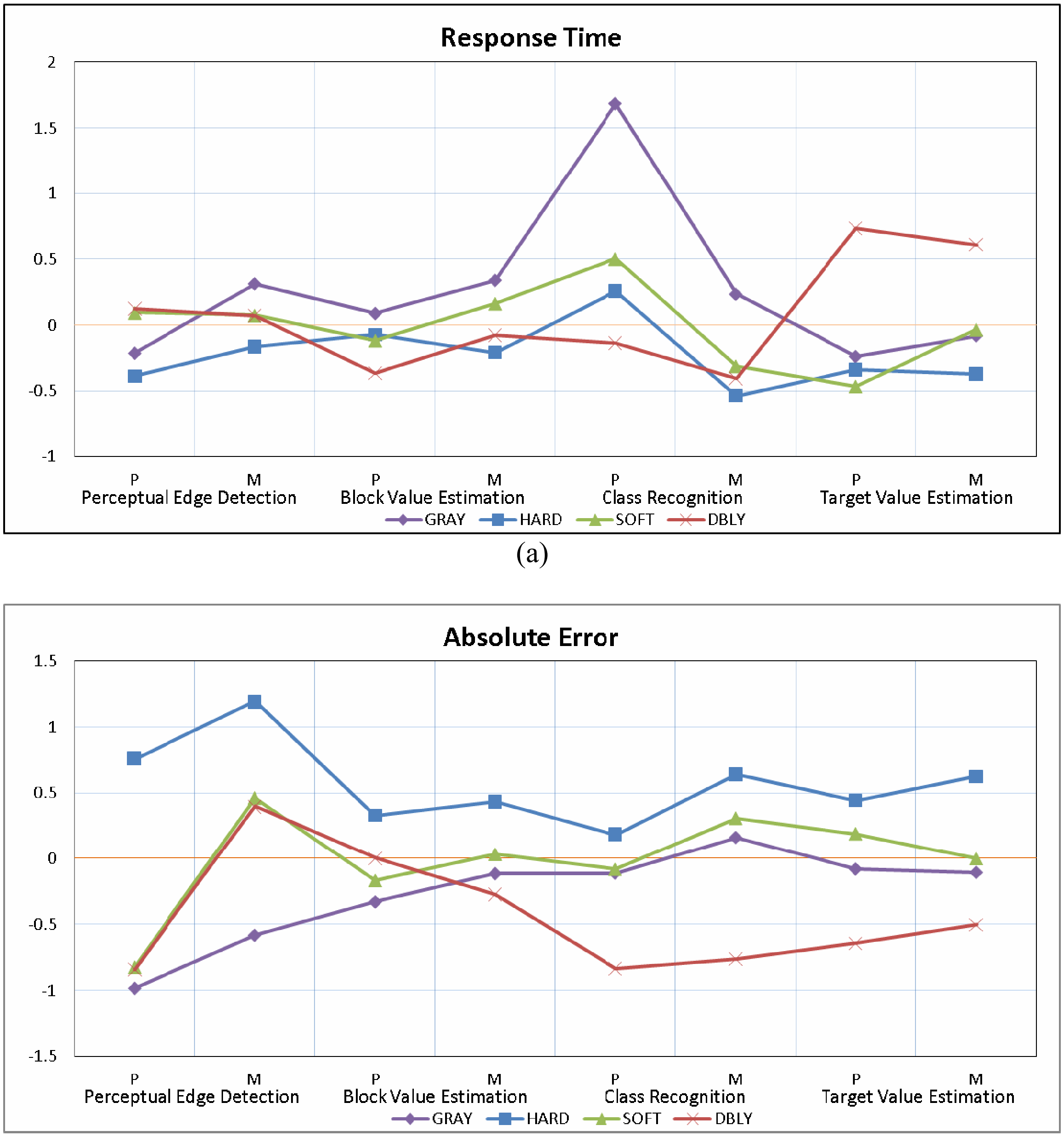
Figure 7.3 The comparison of results from two user studies. (a) *Response time* and *absolute error* for class recognition; (b) *response time* and *absolute error* for target value estimation task.

7.1.2 Further Considerations

Synthetic Datasets vs. Real Datasets

It is still an open question whether a user study should use synthetic datasets or real datasets. Real datasets are from application domains. Thus the results based on real datasets are directly applicable. However, sometimes, it is not easy to find representative datasets. Moreover, it may be difficult to quantitatively describe and measure the features in real datasets. Compared to real datasets, synthetic datasets are easily accessed, where ground truth and features are easily controlled. A problem with a user study is if the results from synthetic datasets adequately measure the performance of visualization techniques.

Figure 7.4 presents the *z-score* for each algorithm across tasks. The *z-score transformation*, also called *standardization*, normalizes a distribution by subtracting the mean and dividing by the standard deviation. Therefore, the *z-score* enables the performances of each method to be measured across tasks. In Figure 7.4, “P” represents the pilot study based on synthetic datasets; “M” represents the main study based on a real dataset. A value of *z-score* above zero means larger error or longer *response time*, and a value below zero means smaller error or faster response.



(a)

(b)

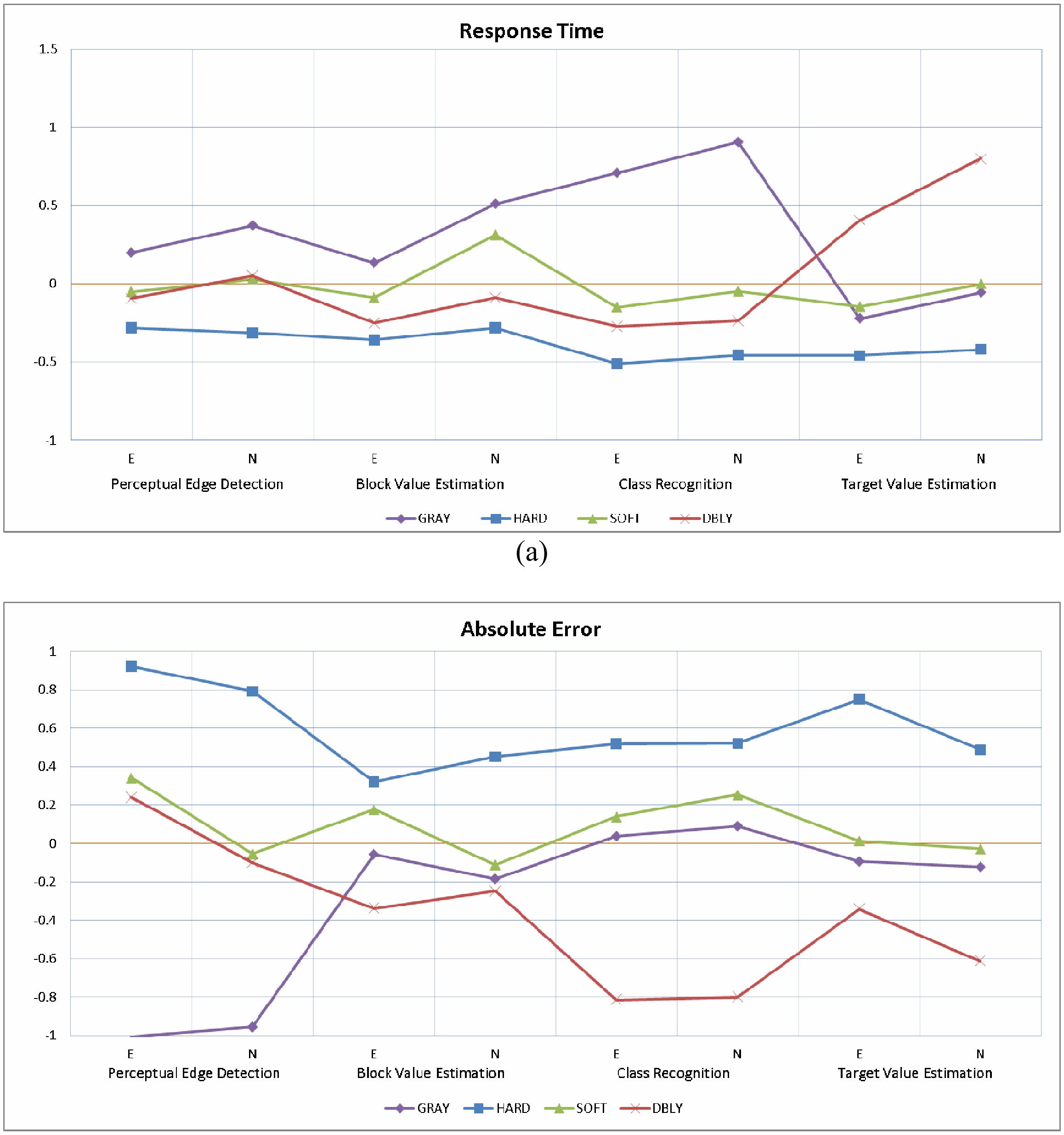
Figure 7.4 The *z-score* for each visualization algorithm calculated for each task. (a) Comparison of *response time*; (b) comparison of *absolute error*. "P" represents the pilot study and "M" represents the main study.

Figure 7.4 does not show any significant difference between each task across the visualization techniques except the jump on the errors of perceptual edge detection from the pilot study to the main study. However the jump on the errors of perceptual edge detection maintains the same order of the techniques in both studies. Based on the comparison of the *z-score*, the synthetic datasets are effective to validate the techniques in our user study.

Expert vs. Non-expert

In the main study, we employed four research associates and two graduate students whose research areas are related to remote sensing. We categorized these six participants as domain experts and the other nine participants as domain non-experts. Figure 7.5 displays the *z-score* comparison for experts and non-experts across all tasks in the main study. Comparing *response time* and *absolute error* indicates there is no significant impact of experts in this study.

However, it does not mean that involving experts in user study is not meaningful. Compared to non-experts, they are domain specialists. They can be more easily trained during the training process. More important, they can propose meaningful requirements on visualizing hyperspectral imagery and provide useful suggestions to improve the techniques.



(a)

(b)

Figure 7.5 The *z-score* comparison between experts and non-experts. (a) Comparison of *response time* for each task; (b) comparison of *absolute error* for each task. “E” represents 6 experts; “N” represents the 9 non-experts.

7.2 Advantages and Limitations of DBLY

7.2.1 Advantages

The DBLY takes advantage of the non-negativity and sum-to-one constraints in a hyperspectral pixel and enhances the detailed local information display capability. The qualitative and quantitative comparisons indicate that adding the DBLY significantly improves the accuracy in visualizing the pixel level and subpixel level information and also has a competitive performance on displaying the global information.

The DBLY is based on LMA, where noise and interference in the original image has been well pre-suppressed. It is particularly useful when the subpixel information is of interest. For example, using the developed visualization system it is easy to estimate the size of some special objects, such as military targets, by just looking at the detail layer. Additional information about an image scene is also provided with interactive exploration tools. Overall, this technique can create informative displays in a succinct form to support practical decision-making.

7.2.2 Limitations

The DBLY scheme blends the pie-chart layer and the background layer. However, the color blending process may create new colors for the image scene. Figure 7.6 displays a sample set of images where a new color is generated by color blending. Figure 7.6(a-b) simulate the two abundance images, where pure red/green column represents a value of 1.0, black column represents 0.0, and mid-level red/green (middle columns in (a) and (b)) represents 0.5. Figure 7.6(c) is the color blending result of Figure 7.6(a) and (b). The middle column of Figure 7.6(c) is very different from red (left column) and green (right column) and is read as a new color by most of observers except color experts. Actually, it is created by blending 50% pure red and 50% pure green. This “new” color may result in a wrong decision when this happens in real applications.

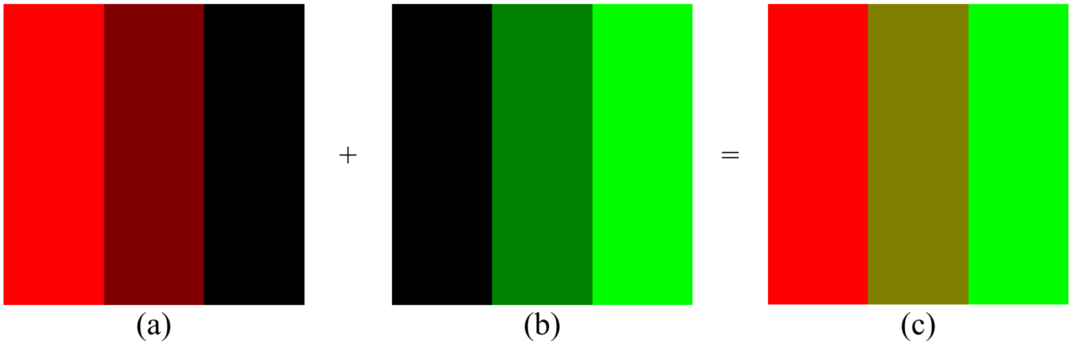


Figure 7.6 An example showing a color blending process where a new color is created.

Our user studies indicated that the *absolute error* of block value estimation is around 14%. The study conducted by Hagh-Shenas [58] indicated that the error of value estimation of blending colors is worse than that of a single color. It may improve the participants’ performance on value estimation if a single color is used to replace a blending color. Icon-based visualization methods may be a potential solution to avoid the color blending. Bokinsky states in her dissertation that icon-based visualization methods can visualize nine variables at most in one image [31].

On the other hand, it may be helpful if the visualized image can directly display the important mixed pixel information without the need of zooming into the subpixel level. For instance, an anomaly is a pixel whose spectral signature is very different from the surrounding pixels, has low probability to be noticed, and a high potential to be a target. If a visualization technique can assist in rapidly “popping out” an anomaly in the mixed pixel composition, it will greatly facilitate the decision-making.

7.3 Conclusion

In summary, this chapter discussed the results of the two studies. The results of the studies demonstrate that our three hypotheses are true. The comparison of the pilot study and the main study shows that the well designed synthetic datasets are suitable for validating the algorithms for visualizing hyperspectral images. The performance of experts and non-experts shows that our studies have a low cognitive level and do not require a strong background to complete. The advantages and disadvantage of using DBLY is also discussed in this chapter. In the following chapter, we will draw the conclusion and present our future work.

CHAPTER VIII

CONCLUSION AND FUTURE WORK

"This is not the end.
It is not even the beginning of the end.
But it is, perhaps, the end of the beginning."
— Winston Leonard Spenser Churchill

In this dissertation, we presented the DBLY visualization strategy and its validation experiments. In this Chapter, we make conclusions about the major aspects of our research and discuss future work. Specially, we introduce a new scheme, multi-layer visualization, which is expected to have better performance.

8.1 Conclusion

1. We have introduced new visualization strategies to visualize hyperspectral imagery. Hyperspectral imagery has many important applications, but how to display the huge amount of information is challenging. In this dissertation, we proposed a new visualization scheme, a double-layer technique, to improve the information rendering. The two user studies revealed that the developed DBLY scheme can display mixed pixel composition and global materials distribution simultaneously.

2. We have designed and conducted two user studies to quantitatively measure the DBLY visualization scheme. We employed user studies to validate the efficiency and weakness of the DBLY visualization algorithm. We conducted a pilot user study based on

synthetic datasets and a main study based on a real dataset. We compared the DBLY visualization algorithm with traditional approaches, such as gray-scale side-by-side display, colored hard classification, and colored soft classification in the two studies. The results show that our scheme has the best performance in displaying local information and has comparable capability of visualizing global information and patterns. The results of the two user studies are consistent.

3. We have added a pie-chart layer to provide high accuracy in conveying local information. The proposed algorithms are the first work that explicitly accounts for the non-negativity and sum-to-one constraints in LMA to visualize hyperspectral imagery. The pie-chart layer, taking advantage of these properties, enhances the detailed local information display capability. Both the pilot and main study prove that adding the pie-chart layer significantly improves the accuracy in visualizing the pixel level and subpixel level information. The pie-chart layer does not degrade the background layer's capability of displaying the global information if a proper blending parameter is used. Our user studies show that the DBLY visualization scheme is competitive to existing visualization techniques at displaying global information.

4. We have implemented a visual exploration tool to visualize hyperspectral imagery. The proposed techniques are implemented in a visualization system. All visualized images presented in this dissertation are screenshots from this visualization system. The interactive tools in a user-friendly interface enable viewers to display an image with any desired level of details.

8.2 Future Work

We do not expect that the proposed algorithms can solve all the problems in hyperspectral image visualization. There is room for further improvement.

Most of the current research visualizes hyperspectral imagery as 2D images. However, compared to 2D visualization, 3D visualization can offer more degrees of freedom and provide more interactive operations. Moreover, the high-spectral-resolution signatures in hyperspectral imagery contain plenty of information. How to directly involve these signatures into the visualization system is a direction of our future work.

The pie-chart layer which takes advantage of the two constrains, i.e., non-negativity and sum-to-one, provides high accuracy in conveying local information. However, many applications may not contain these two constraints. How to extend the pie-chart layer and the developed visualization technique to other application domains needs to be investigated.

Since the pie-charts do not cover the whole image area, the DBLY approach employs another layer to compensate uncovered area. It is relatively complex to control the opacity of the pie-chart layer for displaying proper information. Using a square shaped suppixel may overcome the limitation of the pie-chart. Introducing the other shaped suppixel and reducing the complexity of opacity controlling are the other directions of our future work.

Though the preliminary results and the user studies show that the DBLY technique has the capability to visualize global patterns and local detailed information simultaneously, the DBLY algorithm still has limitations in visualizing hyperspectral images as we discussed in the previous chapter. A multi-layer visualization technique which is presented in next section is expected to overcome the limitations.

8.3 A Potential Solution — Multi-Layer Visualization Scheme

The proposed multi-layer visualization scheme includes five layers. From bottom to top, they are the background layer, the non-uniformed data-driven spots (NUDDS) layers, the pie-chart layer, the oriented sliver layers, and the anomaly layer. In this section, we discuss the multi-layer visualization scheme in detail.

8.3.1 Functions of Five Layers

Five layers are employed to maximize the information that is visualized. Textures with special consideration are utilized to form the final display. Each layer has a specific purpose as described below.

Background Layer: No information is presented by this layer. It is used to enhance the overall appearance of the display. Because it can be seen through the upper layers, the color in the background layer should be carefully chosen to make the final display more appealing. Neutral gray is chosen as suggested in [30][31].

NUDDS layer: This layer displays the statistical distributions of the endmember materials that are widely distributed in the scene (i.e., background materials most likely). For an image with φ background materials, φ *NUDDS* layers are needed with one for each background material. Gaussian-shaped spots were recommended in the original DDS technique [31]. To make it suitable for multiple layer representation, solid circles are used with the radius equal to half of a pixel extent. Moreover, the uniform sampling in the original DDS technique is changed to non-uniform, which is more efficient in capturing subtle variations. We denote this layer as NUDDS. Hence, the fact that spots with the same color are densely packed in a unit area means the corresponding material is more concentrated in this area. The opacity of a spot is controlled by the corresponding abundance in the sampled pixel it represents. In other words, if pixel $\mathbf{r}_{ij}$ is the sampled pixel for the spot at (i, j) in the *k*-th NUDDS sublayer representing the *k*th endmember, the opacity $o_k(i, j)$ is determined as

$$o_k(i, j) = \alpha_k(i, j) \tag{8.1}$$

where $\alpha_k(i, j)$ is the abundance of $\mathbf{m}_k$ in pixel $\mathbf{r}_{ij}$.

Pie-chart layer: This layer is used to display the detailed composition of each sampled pixel. The opacity should be low when visualizing the overall distribution to reduce the line pattern artifacts. Middle or high opacity is more appropriate for the ROI visualization. In addition, the radius of the pie-charts has to be reduced from that used in [23] and distinct from that of the dots in the NUDDS layer to work effectively in the multiple layer situation. The overall opacity of the pie-chart layer is associated with a f parameter.

Oriented Sliver Layer: This layer is to represent anomalous materials (associated with anomalies or targets), which are not spatially well distributed. These materials cannot be well represented by the NUDDS layer because of their low occurrence probability. Thus oriented slivers are used to emphasize these materials as long as they are present in a pixel. Different materials are distinguished by orientation. The opacity of a sliver is controlled by the abundance of the anomalous endmember in the pixel.

Anomaly layer: An anomaly is a potential target. The anomalous pixels should be significantly highlighted. A study by Acevedo *et al.* shows that increasing icon brightness provides better visibility than changing icon size when visualizing scalar values [39]. However, in a very large image scene, even a very bright color cannot make an anomaly spanning only one or two pixels noticeable due to its small size. Hence large 3D icons with bright colors are employed in this layer to represent anomalies.

It should be noted that the colors assigned to endmembers in all layers (NUDDS, pie-chart, oriented slivers, and anomaly) are the same. Layer transparency/opacity can be automatically or manually adjusted.

8.3.2 Material Categorization

The anomalous endmembers are not widely distributed. Under an unsupervised situation, the category of endmember $\mathbf{m}_k$ can be determined by calculating the overall distribution index, I_k, defined as

$$I_k = \frac{M_k N_k}{N^2} \tag{8.2}$$

where M_k is the total distributed amount of endmember $\mathbf{m}_k$, i.e.,

$$M_k = \sum_{i,j} \alpha_k(i, j) \tag{8.3}$$

N_k is the total number of pixels whose maximum abundance is from $\mathbf{m}_k$, and N is the total number of pixels in the scene. If I_k is less than a threshold η_m, $\mathbf{m}_k$ is considered an anomalous endmember; otherwise, it is a non-anomalous endmember. η_m is set to be 1×10^{-3} in this study. An anomalous material will be emphasized by the oriented sliver layer, which can be a foreground material. For a pixel to be considered an anomaly and thus be highlighted in the anomaly layer, the abundance value should be greater than a threshold η_p. In this study, η_p is set to be 0.8~0.9. A non-anomalous endmember is widely distributed and usually a background material. It will be displayed by the NUDSS layer.

8.3.3 Sampling

In Bokinsky's DDS technique, spot density is independent of the scalar field because uniform sampling is used. This makes any subtle variation in distribution unnoticeable. Therefore, non-uniform sampling is proposed in this research. The sampling process for the *k*-th material represented by the *k*-th sublayer has the following steps:

1. Set two controlling parameters D_1 and D_2. The initial sampled pixel set $\Omega = \varnothing$.
2. Randomly choose a pixel $\mathbf{r}_{ij}$ from the image scene.
3. Calculate the threshold η_D for $\mathbf{r}_{ij}$ as:

$$\eta_D(i, j) = (1 - \alpha_k(i, j))(D_1 - D_2) + D_2 \tag{8.4}$$

4. Calculate the coordinate distance between $\mathbf{r}_{ij}$ and each pixel in Ω. If all the distances are greater than η_D, $\mathbf{r}_{ij}$ is added to the sample set Ω.
5. Repeat steps 2 through 4 until the number of iterations are sufficiently large, say, 0.5*N*.

Here, D_1 and D_2 are the minimum distances allowed between two samples when $\alpha_k(i, j)$ takes the smallest (i.e., $\alpha_k(i, j)$ = 0) and the largest value (i.e., $\alpha_k(i, j)$ = 1), respectively. They together control the final spot density. In our experiments $D_1 = 5$ and $D_2 = 1$. Obviously, this sampling algorithm is well correlated with the local abundance of the k-th material. Because the threshold in Eq. (8.4) is smaller for pixels with larger abundance, these pixels have a greater chance of being selected. After the sample points are generated, the opacity of the sample points is set to the abundance value ($\alpha_k(i, j)$).

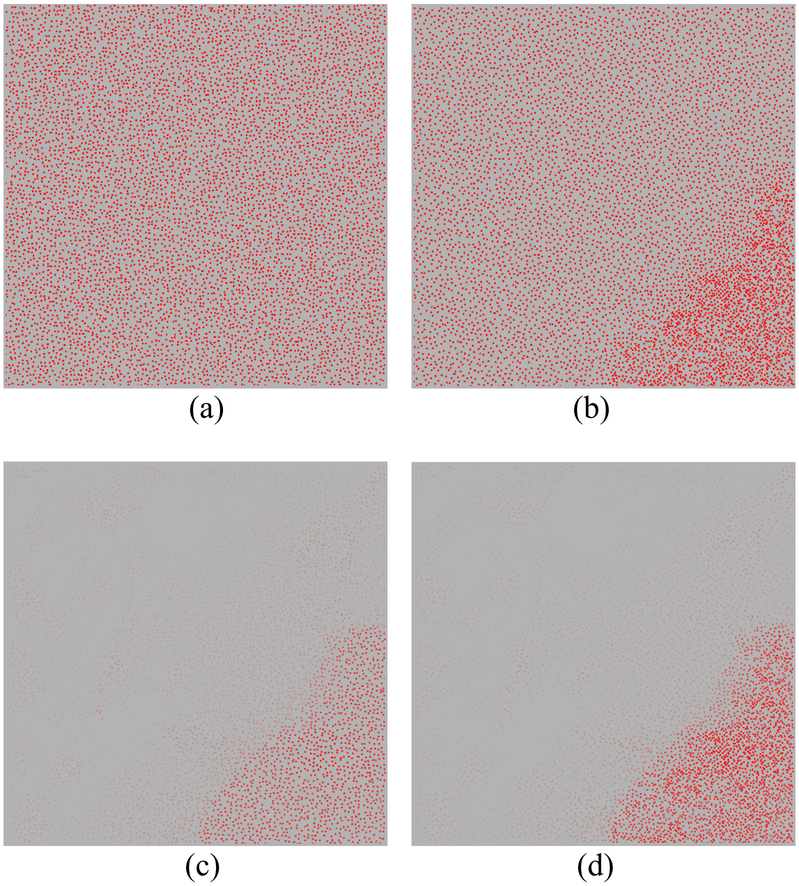


Figure 8.1 The results of two sampling methods. (a) Uniform sampling; (b) non-uniform sampling; (c) uniform sampling after transparency control; (d) non-uniform sampling after opacity control.

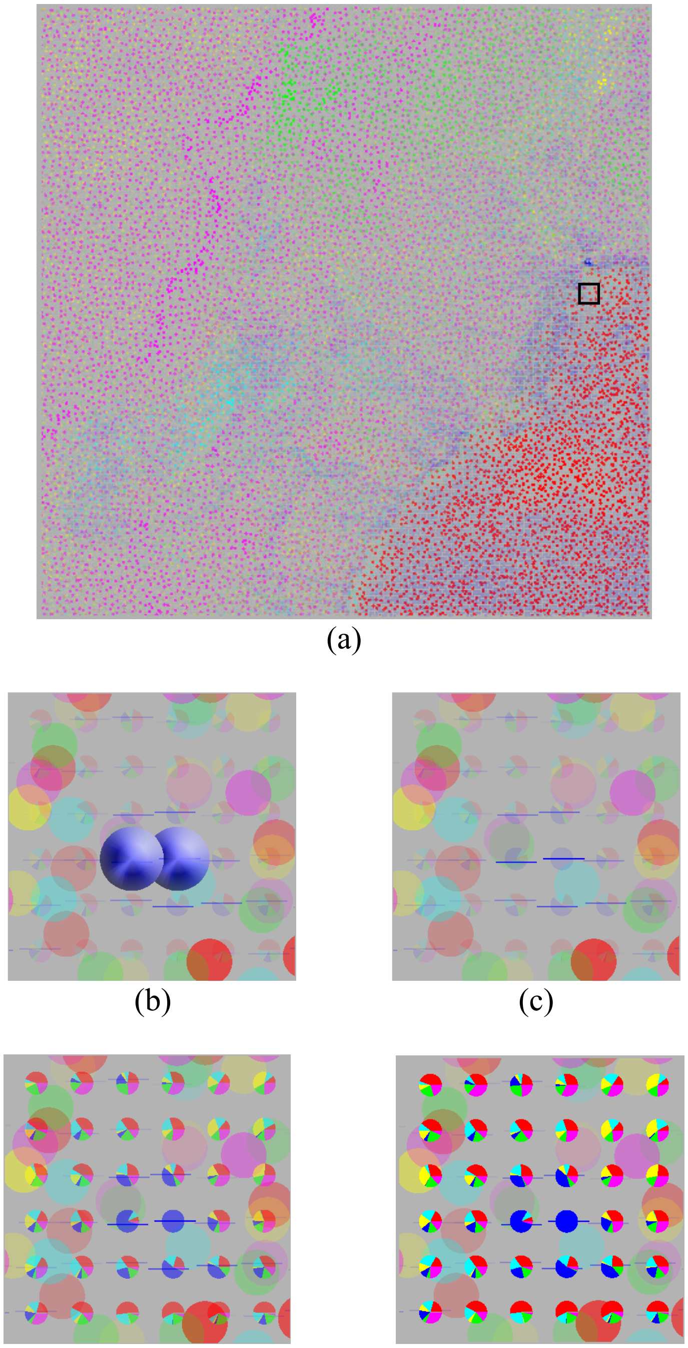
Figure 8.1 shows the sampling results for Playa Lake (the 1st endmember material in Figure 3.4). It is concentrated at the lower-right corner. Figure 8.1 (a) is the (original) uniform sampling result, and Figure 8.1(b) is the non-uniform sampling result where spots at the lower-right corner have higher density. After the opacity/transparency control, Figure 8.1(d) reflects the actual distribution variation of Playa Lake while Figure 8.1(c) does not.

8.3.4 Results of Multi-layer Visualization Scheme

After each layer has been generated, the final display is formed by alpha-blending, a standard computer graphics algorithm for the semi-transparent image display. By default, the blending parameter for the anomaly layer and background layer are 1.0, which means 100% opacity; those for the NUDDS layer and oriented sliver layer are determined by the abundance value in a pixel at (*i, j*); that for the pie-chart is associated with the zooming parameter. It should be noted that the sublayers in the NUDDS layer go through a similar blending process.

The final multi-layer display for the AVIRIS Lunar Lake scene is shown in Figure 8.2. The overall display in Figure 8.2(a) manifests the variations of the six endmember materials, and the anomaly is more visible. Figure 8.2(b) is the ROI with the anomaly layer; the precise location of the anomaly can be easily identified. Figure 8.2(c-e) are the images when the opacity of the pie-chart layer is varied to increase the visibility of the detailed pixel information. In Figure 8.2(c-e), the anomaly layer is deselected. We can also see the role of the oriented sliver layer, which is to make the pixels with the anomalous materials be more easily detected.

Compared to other existing multivariate visualization techniques that cannot provide the precise distribution at subpixel level, the embedded pie-chart layer provides the precise pixel composition. This is particularly useful when a small region is of great interest. For example, it is feasible to estimate the size of small objects, such as invasive species or military targets, based upon material abundance within a pixel. The anomaly layer highlights anomalous pixels, such as small targets, which can greatly facilitate target detection. The NUDDS layer displays the widely distributed endmembers, such as background materials. The NUDDS can reflect the subtle spatial variation in endmember distribution. The oriented silver layer further emphasizes the distribution of important target materials.



(a)

(b) (c)

(d) (e)

Figure 8.2 The multi-layer visualization of AVIRIS Lunar Lake. (a) The overall display with five layers; (b) the ROI in the anomaly layer; (c-e) displays with the opacity of the pie-chart layer being 0.1, 0.5, and 1.0, respectively (without the anomaly layer).

REFERENCES

[1] T. M. Lillesand, R. W. Kiefer, and J. W. Chipman, Eds., *Remote Sensing and Image Interpretation*, 5th edition, Wiley, 2003.

[2] C. Johnson, “Top scientific visualization research problems,” *IEEE Computer Graphics and Applications*, vol. 24, no. 4, pp. 13-17, Jul./Aug. 2004.

[3] “Chesapeake bay and mid-atlantic from space: Three band color composite imagery.” [Online]. Available: http://chesapeake.towson.edu/data

[4] P. K. Robertson and J. F. O’Callaghan, “The application of perceptual color spaces to the display of remotely sensed imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 26, no. 1, pp. 49-59, Jan. 1988.

[5] J. M. Durand and Y. H. Kerr, “An improved decorrelation method for the efficient display of multispectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 27, no. 5, pp. 611-619, May 1989.

[6] J. S. Tyo, A. Konsolakis, D. I. Diersen, and R. C. Olsen, “Principal-components based display strategy for spectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 3, pp. 708-718, Mar. 2003.

[7] V. Tsagaris, V. Anastassopoulos, and G. A. Lampropoulos, “Fusion of hyperspectral data using segmented PCT for color representation and classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 10, pp. 2365-2375, Oct. 2005.

[8] C.-I. Chang and Q. Du, “Interference and noise-adjusted principal components analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2387-2396, Sep. 1999.

[9] S. Cai, Q. Du, R. Moorhead, M. J. Mohammadi-Aragh, and D. Irby, “Noise-adjusted principal component analysis for hyperspectral remotely sensed imagery visualization,” *Proceedings of IEEE Visualization Conference (Compendiu*m), pp. 119-120, 2005.

[10] Q. Du, N. Raksuntorn, S. Cai, and R. Moorhead, “Color display for hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1858-1866, Jun. 2008.

[11] N. P. Jacobson and M. R. Gupta, “Design goals and solutions for display of hyperspectral images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 11, pp. 2684-2693, Nov. 2005.

[12] P. Shippert, “Why use hyperspectral imagery,” *Photogrammetric Engineering & Remote Sensing*, vol. 70, no. 4, pp. 377, 379-380, 2004.

[13] P. Cammpadelli, R. Schettini, and S. Zuffi, “A system for the automatic selection of conspicuous color sets for qualitative data display,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 10, pp. 2283-2286, Oct. 2001.

[14] A. Marcal, “Automatic color indexing of hierarchically structured classified images,” *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, vol. 7, pp. 4976-4979, 2005.

[15] J. J. Settle and N. A. Drake, “Linear mixing and the estimation of ground cover proportions,” *International Journal of Remote Sensing*, vol. 14, no. 6, pp. 1159-1177, 1993.

[16] J. B. Adams, M. O. Smith, and P. E. Johnson, “Spectral mixture modeling: A new analysis of rock and soil types at the Viking Lander 1 site,” *Journal of Geophysical Research*, vol. 91, no. B8, pp. 8098-8112, 1986.

[17] N. Keshava and J. F. Mustard, “Spectral unmixing,” *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44-57, Jan. 2002.

[18] D. M. Rogge, B. Rivard, J. Zhang, and J. Feng, “Iterative spectral unmixing for optimizing per-pixel endmember sets,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 12, pp. 3725-3736, Dec. 2006.

[19] Y. H. Hu, H. B. Lee, and F. L. Scarpace, "Optimal linear spectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 1, pp. 639-644, Jan. 1999.

[20] D. C. Heinz and C.-I. Chang, “Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 529-545, Mar. 2001.

[21] J. Li and L. Bruce, “Wavelet-based feature extraction for improved endmember abundance estimation in linear unmixing of hyperspectral signals,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 3, pp. 644-649, Mar. 2004.

[22] R. Taylor, “Visualizing multiple fields on the same surface,” *IEEE Computer Graphics and Applications*, vol. 22, no. 3, pp. 6-10, May/Jun. 2002.

[23] S. Cai, Q. Du, and R. Moorhead, “Hyperspectral imagery visualization using double layers,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3028-3036, Oct. 2007.

[24] R. Wessels, M. Buchheit, and A. Espesset, “The development of a high performance, high volume distributed hyperspectral processor and display system,” *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, vol. 4, pp. 2519-2521, 2002.

[25] R. Crawfis, *New techniques for the scientific visualization of three-dimensional multi-variate and vector fields*, Ph.D. dissertation, University of California at Davis, 1995.

[26] C. Forsell, S. Seipel, and M. Lind, “Simple 3D glyphs for spatial multivariate data,” *Proceedings of IEEE Information Visualization Symposium*, pp. 119-124, 2005.

[27] C. Weigle, W. Emigh, G. Liu, R. Taylor, J. Enns, and C. Healey, “Oriented sliver textures: A technique for local value estimation of multiple scalar fields,” *Proceedings of Graphics Interface Conference*, pp. 163-170, 2000

[28] C. G. Healey, *Effective visualization of large multidimensional datasets*, Ph.D. dissertation, The University of British Columbia, 1996.

[29] T. Urness, V. Interrante, E. Longmire, I. Marusic, S. O’Neill, and T. W. Jones, “Strategies for the visualization of multiple 2d vector fields,” *IEEE Computer Graphics and Applications*, vol. 26, no. 4, pp. 74-82, Apr. 2006.

[30] R. M. Kirby, H. Marmanis, and D. H. Laidlaw, “Visualizing multivalued data from 2D incompressible flows using concepts from painting,” *Proceedings of IEEE Visualization Conference*, pp. 333-340, 1999.

[31] A. A. Bokinsky, *Multivariate data visualization with data-driven spots*, Ph.D. dissertation, University of North Carolina at Chapel Hill, 2003.

[32] R. J. Vickery, *New visualization techniques for multi-dimensional variables in complex physical domains*, Ph.D. dissertation, Mississippi State University, 2003.

[33] C. A. Bateson, G. P. Asner, and C. A. Wessman, “Endmember bundles: A new approach to incorporating endmember variability into spectral mixture analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 2, pp. 1083-1094, Feb. 2000.

[34] C. M. Schweik and G. M. Green, “The use of spectral mixture analysis to study human incentives, actions, and environmental outcomes,” *Social Science Computer Review*, vol. 17, no. 1, pp. 40-63, 1999.

[35] Q. Du and C.-I. Chang, “Linear mixture analysis-based compression for hyperspectral image analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 4, pp. 875-891, Apr. 2004.

[36] V. Interrante, “Harnessing natural textures for multivariate visualization,” *IEEE Computer Graphics and Applications*, vol. 20, no. 6, pp. 6-11, Nov./Dec. 2000.

[37] G. S. Schmidt, S. L. Chen, A. N. Bryden, M. A. Livingston, L. J. Rosenblum, and B. R. Osborn, “Multidimensional visual representations for underwater environmental uncertainty,” *IEEE Computer Graphics and Application*, vol. 24 no. 5, pp. 56-65, Sep./Oct. 2004.

[38] A. S. Bair, D. H. House, and C. Ware, “Texturing of layered surfaces for optimal viewing,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1125-1132, Sep./Oct. 2006.

[39] D. Acevedo and D. H. Laidlaw, “Participative Quantification of perceptual interactions among some 2D scientific visualization methods,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1133-1140, Sep./Oct. 2006.

[40] H. Levikowitz, “Color icons: merging color and texture perception for integrated visualization of multiple parameters”, *Proceedings of the 2nd Conference on Visualization*, pp.164-170, 1991.

[41] V. Tsagaris and V. Anastassopoulos, “Multispectral image fusion for improved RGB representation based on perceptual attributes”, *International Journal of Remote Sensing*, vol. 26, no. 15, pp. 3241-3254, 2005.

[42] C. Pohl and J. L. V. Genderen, “Multisensor image fusion in remote sensing: concepts, methods and applications”, *International Journal of Remote Sensing*, vol. 19, no. 5, pp. 823-854, 1998.

[43] F. A. Kruse, A. B. Lefkoff, J. W. Boardman, K. B. Heidebrecht, A. T. Shapiro, J. P. Barloon, and A. F. H. Goetz, “The spectral image processing system (SIPS)—Interactive visualization and analysis of imaging spectrometer data”, *Remote Sensing of environment*, vol. 44, no. 2-3, pp. 145-163, 1993.

[44] J. C. Harsanyi and C.-I Chang, “Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach,” *IEEE Transactions on Geoscience Remote Sensing*, vol. 32, no. 4, pp. 779-785, Apr. 1994.

[45] T. Han, D. G. Goodenough, A. Dyk, and J. Love, “Detection and correction of abnormal pixels in Hyperion images,” *Proceedings of IEEE Geoscience and Remote Sensing Symposium,* vol. 3, pp. 1327-1330, 2002.

[46] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd edition, Prentice Hall, 2002.

[47] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, Wiley-Interscience, 2000.

[48] D. H. Brainard, *Colorimetry*, in *OSA Handbook of Optics: Fundamentals, Techniques, and Design*, vol. 1, M. Bass ed., McGraw-Hill, 1995.

[49] C. Ware, *Information Visualization*, 2nd edition, Morgan Kaufmann, 2004.

[50] P. Cammpadelli, R. Schettini, and S. Zuffi, “A system for the automatic selection of conspicuous color sets for qualitative data display,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 10, pp. 2283-2286, Oct. 2001.

[51] D. Hearn and M. P. Baker, *Computer Graphics with OpenGL*, 3rd edition, Prentice Hall, 2004.

[52] D. A. Keim, “Designing pixel-oriented visualization techniques: theory and application,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 59-78, Jan./Feb. 2000.

[53] J. Alello and L. Korn, “MGV: a system for visualizing massive multidigraphs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 1, pp 21-38, Jan./Feb. 2002.

[54] C. G. Healey, “Choosing effective colours for data visualization,” *Proceedings* of *IEEE Visualization Conference*, pp. 263-270, 1996.

[55] D. H. Laidlaw, R. M. Kirby, J. S. Davidson, T. S. Miller, M. Silva, W. H. Warren, and M. Tarr, “Quantitative comparative evaluation of 2D vector field visualization methods”, *Proceedings of IEEE Visualization Conference*, pp. 143-150, 2001.

[56] M. O. Ward and K. J. Theroux, “Perceptual benchmarking for multivariate data visualization,” *Proceedings of IEEE Visualization Conference*, pp. 314-321, 1997.

[57] R. Kosara, C. G. Healey, W. Interrante, D. H. Laidlaw, and C. Ware, “User studies: why, how, and when?”, *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 20-25, Jul./Aug. 2003.

[58] H. Hagh-Shenas, S. Kim, V. Interrante, and C. Healey, “Weaving versus blending: a quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1270-1277, Nov./Dec. 2007.

[59] D. C. Howell, *Statistical Methods for Psychology*, 5th edition, Duxbury, 2002.

[60] V. Barnett and T. Lewis, *Outliers in Statistical Data*, 3rd edition, John Wiley and Sons, 1994.

[61] R. Azuma and C. Furmanski, “Evaluating label placement for augmented reality view management,” *Proceedings* of *Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 66-75, 2003.

[62] L. Bastin, J. Wood, and P. F. Fisher, “Visualising and tracking uncertainty in thematic classifications of satellite imagery,” *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, vol. 5, pp. 2501-2503, 1999.

[63] J. L. Dungan, D. Kao, and A. Pang, “The uncertainty visualization problem in remote sensing analysis,” *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, vol. 2, pp. 729-731, 2002.

[64] M. B. Gousie and S. Milewski, “A system for 3d error visualization and assessment of digital elevation models,” *Proceedings of IEEE International Geoscience and Remote Sensing Symposium,* pp. 4064-4067, 2007.

[65] H. Li, C.-W. Fu, Y. Li, and A. J. Hanson, “Visualizing large-scale uncertainty in astrophysical data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1640–1647, Nov./Dec. 2007.

[66] S. K. Lodha, A. Pang, R. E. Sheehan, and C. M. Wittenbrink, “Uflow: visualizing uncertainty in fluid flow,” *Proceedings of IEEE Visualization Conference*, pp. 249-254, 1996.

[67] C. Lundstrom, P. Ljung, A. Persson, and A. Ynnerman, “Uncertainty visualization in medical volume rendering using probabilistic animation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1648-1655, Nov./Dec. 2007.

[68] A. Doi and K. Koyamada, “Error visualization of tetrahedral subdivision approach for trilinear interpolation,” *Proceedings of Parallel and Distributed Systems*: *Workshops, Seventh International Conference*, pp. 195-200, 2000.

[69] G. Grigoryan and P. Rheingans, “Point-based probabilistic surfaces to show surface uncertainty,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 5, pp. 564-573, Sep./Oct. 2004.

[70] S. Haroz and K. Heitmann, “Seeing the difference between cosmological simulations,” *IEEE Computer Graphics and Applications*, vol. 28, no. 5, pp. 37-45, Sep./Oct. 2008.

[71] S. A. Bly and J. K. Rosenberg, “A comparison of tiled and overlapping windows,” *Proceedings of Human Factors in Computing Systems Conference*, vol. 17, no. 4, pp. 101-106, 1986.

[72] S. Biffl, B. Thurnher, G. Goluch, D. Winkler, W. Aigner, and S. Miksch, “An empirical investigation on the visualization of temporal uncertainties in software engineering project planning,” *Proceedings of International Symposium on Empirical Software Engineering*, pp. 437-446, 2005.

[73] R. Pausch, M. A. Shackelford, and D. Proffitt, “A user study comparing headmounted and stationary displays,” *Proceedings of IEEE Symposium on Research Frontiers in Virtual Reality*, pp. 41-45, 1993.

[74] J. P. Martin, J. E. Swan, R. J. Moorhead, Z. Liu, and S. Cai, “Results of a user study on 2D hurricane visualization,” *Computer Graphics Forum*, vol. 27, no. 3, pp. 991-998, 2008.

[75] A. M. Lopes, *Accuracy in scientific visualization*, Ph.D. dissertation, University of Leeds, 1999.

[76] Z. Liu, J. E. Swan, S. Cai, R. J. Moorhead, and J. P. Martin, “Using explicit flow synthesis and implicit task design for objective flow visualization user study,” *Technical Report #7002,* GeoResources Institute, Mississippi State University, pp. 1-26, 2008.

[77] M. Cochrane, “Using vegetation reflectance variability for species level classification of hyperspectral data,” *International Journal of Remote Sensing*, vol. 21, no. 10, pp. 2075-2087, 2000.

[78] N. C. Coops, C. Stone, R. N. Merton, L. Chisholm, “Assessing Eucalypt foliage health with field-based spectra and high spatial resolution hyperspectral imagery,” *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, pp. 603-605, 2001.

[79] T. Smith and J. Guild, “The C.I.E. colorimetric standards and their use,” *Transactions of the Optical Society*, vol. 33, no. 33, pp. 73-134, 1932.