A DATA FLOW APPROACH TO COLOR GAMUT VISUALIZATION

by

CHAD ANDREW ROBERTSON

A THESIS

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

June 1997

A DATA FLOW APPROACH TO COLOR GAMUT VISUALIZATION

by

CHAD ANDREW ROBERTSON

A THESIS

Presented to the Department of Computer and Information Science
and the Graduate School of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

June 1997

"A Data Flow Approach to Color Gamut Visualization," a thesis prepared by Chad A. Robertson in partial fulfillment of the requirement for the Master of Science degree in the Department of Computer and Information Science. This thesis has been approved and accepted by:

Gary W. Meyer

Gary W. Meyer, Thesis Advisor

6/3/97

Date

Accepted by:

Vice Provost and Dean of the Graduate School

An Abstract of the Thesis of

Chad Andrew Robertson for the degree of Master of Science
in the Department of Computer and Information Science to be taken June 1997

Title: A DATA FLOW APPROACH TO COLOR GAMUT VISUALIZATION

Approved: Gary W. Meyer
Dr. Gary W. Meyer

Computer software has been created to assist color reproduction engineers in the evaluation of color printing equipment. The software system uses scientific visualization techniques to allow engineers to view and interact with color data. The system is based upon a modular data flow architecture. Color gamut visualization applications are constructed by connecting data flow modules in a visual environment to build application networks. Modules have been programmed for reading and generating color data, transforming color data, and displaying color data. Applications can incorporate advanced visualization techniques to simultaneously analyze multiple color data sets. New modules can be added to the system to support future developments in color reproduction and scientific visualization.

CURRICULUM VITA

NAME OF AUTHOR: Chad Andrew Robertson

PLACE OF BIRTH: Long Beach, California

DATE OF BIRTH: March 28, 1969

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon
Virginia Polytechnic Institute and State University
James Madison University
University of Arizona

DEGREES AWARDED:

Master of Science in Computer Science, 1997, University of Oregon
Bachelor of Science in Computer Science, 1991, James Madison University

AREAS OF SPECIAL INTEREST:

3-D Computer Graphics, Human Computer Interaction

PROFESSIONAL EXPERIENCE

Research Assistant, Department of Computer and Information Science, University of Oregon, Eugene, 1996-1997

Computer Scientist, Department of Defense - Naval Space Command, Dahlgren, Virginia, 1992-1995

ACKNOWLEDGEMENTS

The author wishes to express his sincere thanks and appreciation to Dr. Gary Meyer for his guidance and assistance throughout the research and preparation of this thesis. His support and enthusiasm provided the focus for what has been a stimulating and rewarding project.

This research was funded by a grant from the Hewlett Packard Company.

DEDICATION

To Teri, for the opportunity and purpose.

TABLE OF CONTENTS

Chapter Page

Page

LIST OF FIGURES

Figure Page

Page

CHAPTER I

INTRODUCTION

A computer software visualization tool can aid a color reproduction engineer in analyzing the color output capabilities of new hardware devices. A necessary step in the design of color reproduction equipment is the determination of the location and size of the color gamut for the machine [13]. The color gamut of a device is the three dimensional volume of all colors, plotted in a color space, that the hardware's primaries can reproduce. Because a color gamut exists as a three dimensional object, the use of computer graphics to visualize its size and shape is appropriate [14].

Software visualization tools employ computer graphics to present data in meaningful ways to an analyst. The visualization of this data, as a method of interpretation for an engineer, makes these tools useful in many steps of the design of color equipment. An engineer might want to analyze the gamut of a potential device with a specific set of primaries; for example, inks in a color printer or phosphors in a color monitor. This is generally done with the objective of maximizing the size of the gamut to distribute colors uniformly through the color space [13]. A graphical representation of this gamut makes it possible to compare the output of different sets of primaries. In another situation, the engineer may want to analyze measured data representing machines that have already been produced and the impact different conditions may have on their output. For instance, measured data for ink fade sets can be used with the visualization software to show the effects of fading or aging on the ink primaries of a color printer.

There are additional factors that make visualization tools good choices for color gamut visualization. Interactive computer graphic programs that produce colorimetrically correct three dimensional pictures are difficult to develop from scratch and their ability to be modified and extended is limited. Commercially available visualization tools provide a framework for building such applications. They include facilities for accurate graphical visualizations of data in the context of a data flow architecture. The tools allow for interactive control over the processing of the analyzed information [14]. Additionally, the modularity of the architecture offers ease of extensibility.

The motivation for the work that is described in this thesis was to develop a software tool that can be used by color reproduction engineers in the design and analysis of color output devices. These devices are mainly color printers, but may also include other color input and output equipment such as monitors and scanners. A second motivation was to create a tool that will serve as a teaching aid to help researchers and students in learning about the concepts and principles involved in color science. A goal was to provide an interactive visual supplement to the current literature on the subject. With the three dimensional computer graphic visualizations the tool provides, users will be able to examine color spaces and manipulate them directly. It is hoped that the result will increase the speed with which color science principles are learned.

The color gamut visualization system merges concepts from two distinct fields of study. It illustrates many aspects of color science through the use of scientific visualization practices. The result is a software tool that incorporates fundamentals of both areas to achieve a solution to color reproduction problems. The tool is designed to grow as advances in either field are made. Extensions in functionality will allow the system to stay up to date with new developments in color science and scientific visualization.

In preparation for a description of the color gamut visualization system, this thesis begins with two chapters covering background information. The first discusses fundamentals of color science. This chapter gives the information necessary for understanding many of the principles that are found and implemented in the color gamut visualization system. It begins with a brief historical perspective and continues with descriptions of color spaces and color space transformations. The next chapter deals with scientific visualization and computer graphics. It presents basic concepts and techniques that are used to visualize data with computer graphics. The information is given in the context of methods that are appropriate to the color gamut visualization system. Following these chapters, Chapter IV gives a description of the design and use of the color gamut visualization system. It describes details of the system's architecture and the components it provides to build gamut visualization applications. Included are several example applications that utilize the full functionality of the system.

CHAPTER II

COLOR SCIENCE PRIMER

This chapter will define some of the principles and practices of color science in order to provide a knowledge base for understanding the remaining material presented in this thesis. It is not intended to be a rigorous treatise on color science, but a basic starting point that can serve as an introduction to the study of this subject. The information presented is drawn from a literature survey that was used in the development of this project.

Light and the Spectrum

Sir Isaac Newton (1642 - 1727) is credited with the breakthrough that allowed for the understanding of color to be brought into the modern era. Prior to Newton there was a basic awareness that the colors of objects were related to light and properties of light such as reflection and refraction, but there existed little understanding of color itself. Newton's famous experiment with sunlight and prisms revealed that a beam of white light could be separated into component colors. These colors represent what he called a spectrum. He identified seven component color regions and defined them as red, orange, yellow, green, blue, indigo, and violet. He also made note of the fact that the spectrum was, in reality, made up of a large number of intermediate colors in a smooth transition between each spectral region. The significance of Newton's discoveries was that future attention in the study of color would be directed at the light source, thereby clarifying that color was an identifiable property of light [7, 10]. From Newton, it became clear that without light there exists no color.

The discoveries that lead to our understanding of light as a form of electromagnetic energy were made by J.C. Maxwell (1831 - 1879). Maxwell provided the theory to clearly explain the results of Newton's experiments. An electromagnetic wave, such as light, X-ray, radar, and radio waves, can be characterized by its wavelength [7]. Wavelength is the distance from crest to crest of the waves that make up the electromagnetic energy. For light, the wavelengths are measured in terms of nanometers (nm). While the exact range of light that is visible to the human eye varies slightly from person to person, generally the visible range of light falls between 400nm and 700nm. In Newton's experiments the prism separated the light rays by wavelength. The specific wavelength of the rays caused them to be refracted: more for shorter wavelengths and less for longer wavelengths. The importance of this fact is that it allows for specific numbers to be assigned to points in the visible spectrum, rather than Newton's broad color regions [10]. Figure 1 shows the visible spectrum, indicating the main color regions and their respective wavelengths.

Emission and Absorption

In nature, there are only two ways that objects can affect the wavelength composition of light: either by emitting or absorbing it. An emitter of light, by some means, converts other forms of energy into light energy. An absorber of light has the opposite effect; it converts light energy to some other form.

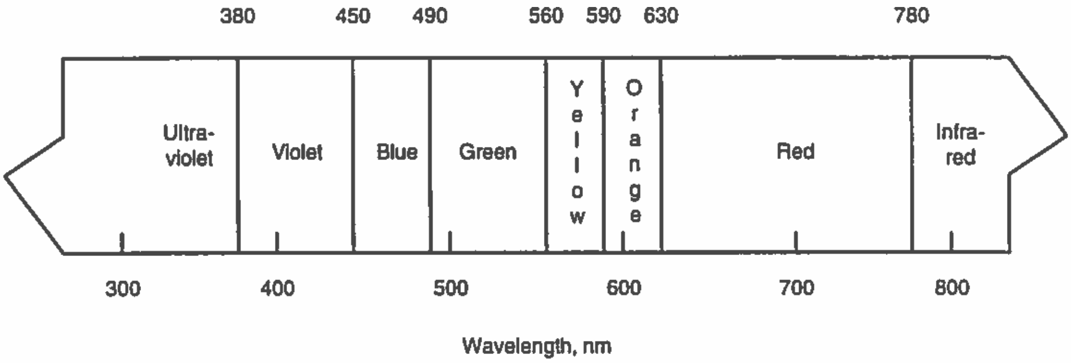

Figure 1. Main bands of color in the spectrum shown against a scale of the wavelength of the light.

Emitters create light through energy transformations. An example of an emitter is an incandescent light bulb, where electrical energy is transformed to light (and thermal) energy. Each energy transformation causes light to be emitted in different amounts at each wavelength through the visible range. For instance, a truly pure white light source, if it existed, would emit light in equal amounts at all wavelengths [10]. It is the difference in wavelength composition that allows for characterization of light sources. This can be seen in the differences in wavelength composition of daylight compared to that of a fluorescent bulb and will be covered in the next section which addresses spectral data.

Absorbers also work by transforming energy. Depending on the composition of the object, light is absorbed and transformed in varying amounts by wavelength. Absorptive objects can be classified as either reflectors or transmitters. A reflector absorbs and re-emits light from its surface, giving the re-emitted light a different wavelength composition (excepting a perfect mirror, which would have a re-emitted wavelength composition identical to the incoming light). A transmitter, such as water, allows light energy to pass through the object such that the light that emerges from the material is also different in wavelength composition. This difference can be based on both the material's composition and thickness.

The wavelength composition of light after interacting with absorptive objects, and after interpretation by the receptors of the eye, give objects their perceived color. The role of emitted and absorbed light in relation to color is among the most fundamental principles of color science.

Spectral Data

Using the knowledge that all objects in nature fall into the categories of emitter, reflective absorber, or transmissive absorber, a next step in identifying and understanding the way light interacts with objects in the world is to describe spectral curves. Spectral curves and the data they represent are the "visual fingerprint" of an object [10]. A spectral curve describes how the object affects light on a wavelength by wavelength basis.

For a reflector, it is possible to graph the reflectance of the object, which is the intensity of reflected light as a percentage of the incoming light at each wavelength in the visible spectrum. Figure 2 shows an example spectral curve for a red colored object. Similarly, for a transmitter, it is possible to graph the object's transmittance, which is a measurement of the percentage of the incoming light that is transmitted through the object.

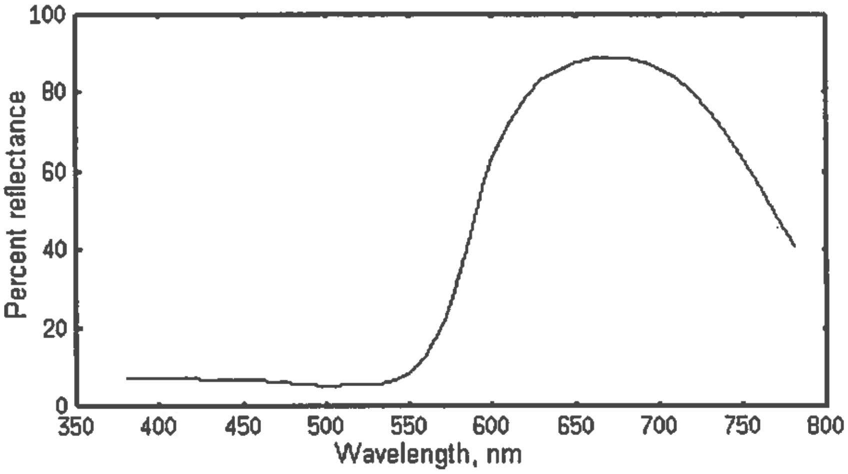

Figure 2. Simulated spectral curve for a red colored object. The object reflects very few shorter wavelengths and almost all of the longer wavelengths.

Emitters also have spectral curves that can be graphed. For an emitter the spectral curve represents the emitted light energy at each wavelength in the visible spectrum relative to the total light energy. This is known as the light's spectral energy distribution. Figure 3 shows the spectral curves for two different light sources: daylight (after interacting with the earth's atmosphere) and a fluorescent light bulb.

Because light is a mixture of all wavelengths and the data that a spectral curve represents can potentially cover all wavelengths in the visible spectrum, spectral data is a complete description of the color information of an object [10]. The emissive properties of a light source determine the wavelength composition of the light. When reflected or transmitted by an object, the wavelength composition is changed. The result is the complete color description of the object. Unfortunately, the information in spectral data does little to describe how the human eye interprets it.

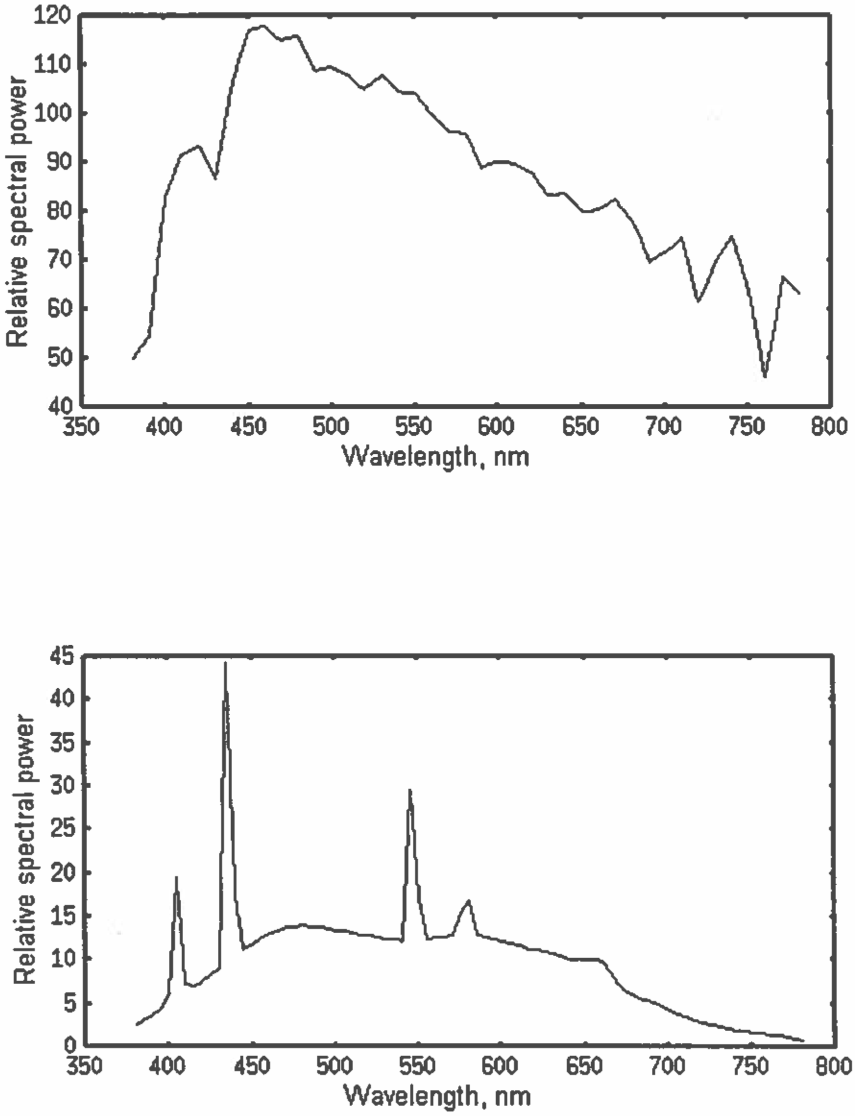

Figure 3. Spectral energy distributions for daylight (top) and a broad-band fluorescent light bulb (bottom).

The Eye and Tristimulus Data

The human eye senses color by using three types of receptors, referred to as cones. One type of cone is most sensitive to the long wavelength region in the visible spectrum, or what is commonly referred to as the "red" region, another to the middle wavelength region, or "green" region, and the last to the short wavelength region, or "blue" region [8, 9]. A plot of the sensitivities of the three types of cones is given in Figure 4.

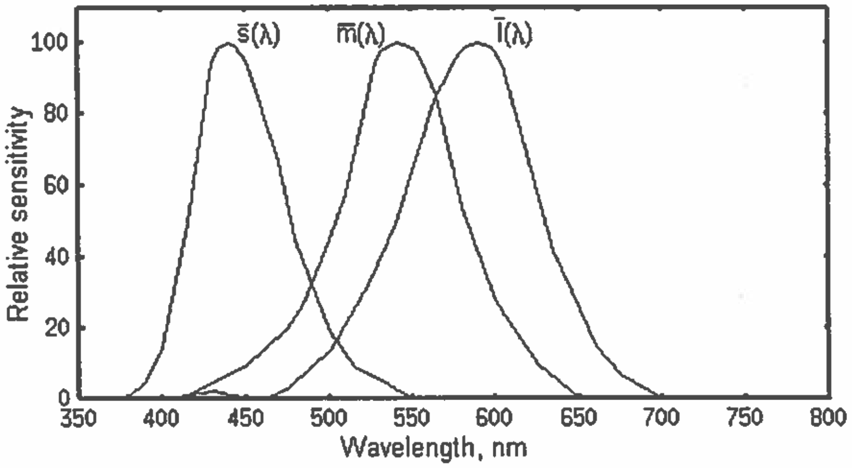

Figure 4. Approximation of the spectral curves defining the sensitivities of the short, medium, and long wavelength receptors of the human eye. Curves are plotted short, medium, and long, left to right.

Understanding the means of color interpretation within the eye leads to tristimulus descriptions of color. A tristimulus description of a color defines the color in terms of three quantities, usually referred to as primaries or stimuli [10]. Color description and measurement in almost every form is tristimulus because of the human eye. The exception is the previously described spectral data description of color.

The values in a tristimulus color description can be calculated directly from spectral data. The SML tristimulus color description is based on the fundamental spectral sensitivities of the short, medium, and long wavelength receptors of the human eye, as depicted in Figure 4. The conversion from spectral information to tristimulus values is done with integration functions. The functions are evaluated at each wavelength across the visible range of the spectrum. They are expressed as:

$$S = \int E(\lambda)\bar{s}(\lambda)d\lambda$$

$$M = \int E(\lambda)\bar{m}(\lambda)d\lambda$$

$$L = \int E(\lambda)\bar{l}(\lambda)d\lambda$$

where $E(\lambda)$ defines the spectral power at wavelength λ, and $\bar{s}(\lambda)$, $\bar{m}(\lambda)$, and $\bar{l}(\lambda)$ are the relative sensitivities for each of the three types of receptors. The spectral power at a wavelength is the product of the spectral energy of the light source and the reflectance (or transmittance) factor at the wavelength [7]. Often, the equations will appear with the term $E(\lambda)$ replaced with separate variables representing the two components that define it.

Two colors that have different spectral power distributions yet match one another are called metamers. With metamers, the tristimulus values which result from integrating each color's spectral composition are the same [9]. This concept is of practical importance in all color reproduction. In many cases, it allows color reproduction engineers to only be concerned with finding metameric matches. However, metameric matches often do not hold when a different observer is used or the illuminant is changed

[7]. Engineers must therefore also understand the conditions that must be present for the match to hold.

The most common tristimulus description of color is RGB. Red, green, and blue are additive primaries. Theoretically, this means that the appearance of any color can be simulated by starting with black, representing no light, and adding proportions of red, green, and blue light to achieve the desired color. With black representing no light, applying equal amounts of each primary at maximum intensity results in white light. RGB exists as the most commonly known tristimulus color description because it presents a good model for designing mass-producible devices that either imitate the eye, such as a scanner, or simulate the reproduction of many colors for the eye to perceive, such as a computer monitor or television [10]. With a scanner, color vision is imitated by the measurement of red, green, and blue light from a sample. A monitor simulates colors by combining varying intensities of red, green, and blue phosphors.

Primaries in a tristimulus color description are not required to be additive. The CMY tristimulus color description uses subtractive primaries. CMY addresses the situation when colors are created starting from white, rather than black as with RGB. This situation is the case with color printing. Conceptually, color printing is an operation of subtracting certain wavelengths from the white of a piece of paper through the use of a filtering pigment (ink). For instance, to subtract red from white, a filter is needed that allows all colors to pass through except red, thereby producing a color with complete absence of red. The filter to eliminate red is colored cyan. Similarly, the filter to eliminate green is magenta, and the filter to eliminate blue is yellow. CMY works analogously to

RGB in that combinations of the primaries, in this case subtractive primaries, can produce almost any perceivable color.

Tristimulus descriptions of color have both advantages and disadvantages. A disadvantage of tristimulus color systems such as RGB is that the precise set of wavelengths that each of the three detectors or emitters in the system sense or reproduce can depend on the device, and is therefore often not consistent from one to another. Another disadvantage is that the tristimulus description of the color of an object depends not only on the properties of the object, but also the light source illuminating it. As previously described, this is not the case with a spectral data description of an object, which allows the object's emissive, reflective, or transmissive properties to be captured separately, giving a color description that is independent of the illuminating light source. Two important advantages of tristimulus color descriptions, however, are that it models human vision, and that it allows for colors to be plotted, or graphed, in three dimensions. The idea of plotting colors in three dimensions is the start for a discussion of color spaces.

Color Spaces and the CIE

A tristimulus color description makes it possible for colors to be plotted in a three dimensional coordinate system. Every color can be represented by a unique point in this system by interpreting the amounts of each of the stimuli, or primaries, relative to a set of three axes. The space that is defined by the three axes that makes up the coordinate system is referred to as a color space [10]. Different color spaces exist, based on different methods of tristimulus color description. The three that have been discussed so far are

SML, RGB, and CMY. Colors can be transformed from one color space to another using a transformation function. Because of this ability to transform between spaces, most any primary system can be used.

Previously it was stated that tristimulus color systems, specifically RGB, can have problems because every device, whether it is the human eye, a scanner, monitor, or printer, has a different definition of what makes up the RGB color space. The difference is due to device dependent implementation of the tristimulus receptors, emitters, or transmitters. An organization of color scientists known as the CIE (in French, the Commission Internationale de l'Éclairage) attempted to solve this problem in 1931 by devising the first of many new tristimulus color systems, a system which uses X, Y, and Z as its values rather than R, G, and B [5, 7]. The meeting of the CIE in 1931 was one of the first comprehensive attempts to devise a system that completely specified the lighting and viewing conditions under which a color could be measured or viewed [10]. In addition to the XYZ color space, among other things, a specification of the Standard Observer was made, which was an attempt to define the average human observer. Also, a set of specifications was created for certain light sources, known as standard illuminants, to be used for color comparison. 1976 was also a year of significant developments and specifications from the CIE. Discussion of select CIE developments follows.

CIE XYZ Color Space

The CIE XYZ color space defines all colors in terms of three primaries X, Y, and Z. These primaries were defined by the CIE to have a number of significant properties.

The XYZ primaries were based on data from human color matching experiments and can therefore accurately predict when a viewer would consider two colors to match. They work like RGB in that they are additive primaries, so every color is a mixture of varying amounts of X, Y, and Z. The Y value in a XYZ tristimulus color description represents the luminance of the color as a function of wavelength. Also important is that in XYZ color space all color values in the space are positive [7]. This means that a color matching experiment using CIE XYZ will not produce matches requiring a negative quantity of one of the primaries, as is possible with RGB. A negative quantity of a primary is required when, to produce a match between combinations of the three RGB primaries and a target color, one of the primaries must be added to the target color rather than the matching color. The CIE, in its definition of the XYZ color space, eliminates this potential for negative values.

The XYZ color space is the foundation for the CIE system. It is part of the 1931 Standard Observer specification as a set of color matching functions that represent how the XYZ tristimulus primaries can be combined to produce all colors in the spectrum [7]. These matching functions are plotted in Figure 5.

XYZ tristimulus values can be calculated from spectral information in a manner similar to SML values. The functions to do this are:

$$X = k\int S(\lambda)\beta(\lambda)\bar{x}(\lambda)d\lambda$$

$$Y = k\int S(\lambda)\beta(\lambda)\bar{y}(\lambda)d\lambda$$

$$Z = k\int S(\lambda)\beta(\lambda)\bar{z}(\lambda)d\lambda$$

where $S(\lambda)$ is the relative power of the illuminant at wavelength λ, $\beta(\lambda)$ is the spectral reflectance or transmittance at wavelength λ, and $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$ are values from the CIE color matching functions. The value of k is constant. It is a value that results in Y being equal to 100 for the perfect reflector or transmitter [7]. The function that defines k is:

$$k = \frac{100}{\int S(\lambda)\bar{y}(\lambda)d\lambda}.$$

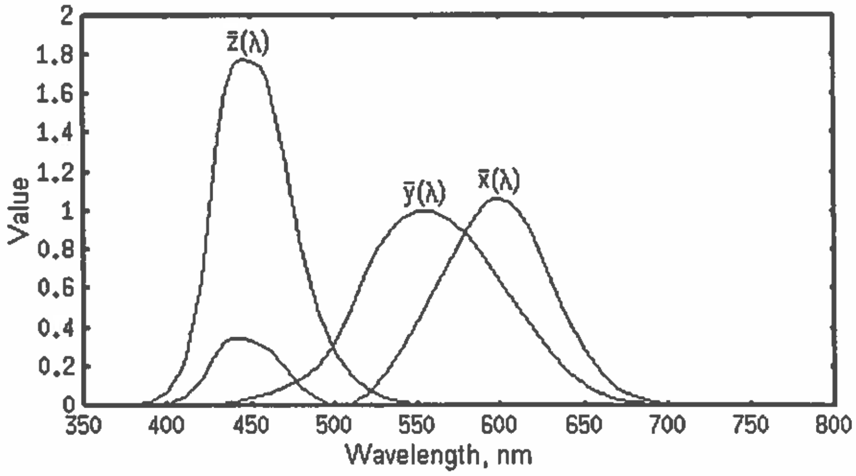

Figure 5. The CIE color matching functions for the 1931 Standard Colorimetric Observer.

CIE xyY Color Space and Chromaticity Diagram

The CIE xyY color space is derived directly from XYZ. It is useful in mapping colors into a two dimensional space that represents them independent of lightness [7]. The Y value encodes the luminance of the color, while x and y are called its chromaticity coordinates. x and y are derived from XYZ and represent the strictly color related, or chromatic, information of the color. Therefore, two colors that are identical except for luminance will have the same chromaticity coordinates. The chromaticity coordinates, which include z, represent the relative magnitude of tristimulus values [9]. Chromaticity coordinates are defined with the following equations:

$$x = \frac{X}{X+Y+Z}$$

$$y = \frac{Y}{X+Y+Z}$$

$$z = 1 - x - y\,.$$

Values in the xyY color space lend themselves to being plotted in a two dimensional graph known as the CIE chromaticity diagram. This diagram is a map of the range of visible colors. There are two components that make up the boundaries of visible colors in the chromaticity diagram. The spectrum locus is a horse-shoe shaped line that is defined by the x and y chromaticity coordinates of the pure wavelengths of the visible spectrum. The plotted wavelengths generally range from 450nm to 700nm. Due to the fact that all visible colors are made up of combinations of these pure wavelengths, all the visible colors of the spectrum are located within the boundary of the curve [10]. The other boundary is the line formed by connecting the endpoints of the spectrum locus. This line is called the purple boundary. Points on the purple boundary define colors that are mixtures of red and blue (or violet depending on the spectrum locus range) as these represent the ends of the spectrum [7]. A version of the CIE chromaticity diagram can be seen in Figure 6.

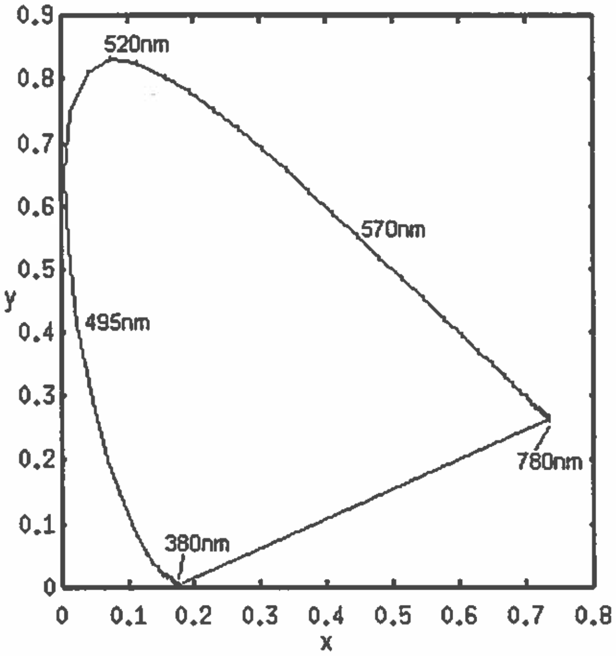

Figure 6. The CIE chromaticity diagram plotted with chromaticity values from 380 to 780 nanometers.

Uniform Color Spaces

In 1976, the CIE created several color spaces which attempt to map colors in a perceptually uniform manner. A uniform color space is one where the distance in the space between two colors corresponds to an observer's judgment of the perceived "closeness" of the two colors [10]. A chromaticity diagram addresses only the proportions of tristimulus values, and not their magnitudes, due to the absence of luminance information. However, colors usually differ in both luminance and chromaticity. The L* primary of the CIE L*u*v*, CIE L*a*b*, and CIE L*C*H* color spaces transforms the luminance value of a color onto a uniform lightness scale. The uniform lightness scale defines lightness values from black to white. In this way, these color spaces allow representation of a color's lightness, while also mapping their chromaticity in perceptually uniform ways.

Uniformity in the L*u*v* color space is achieved by treating u* and v* as mathematically transformed versions of chromaticity coordinates, similar to x and y in the xyY color system [7]. CIE L*u*v*, due to this increased perceptual uniformity over xyY and its representation of a color's luminance, is widely used in industries involving emissive products such as televisions or computer monitors.

Somewhat similarly, a* and b* in the L*a*b* color space are also attempts to represent the chromaticity values of a color on uniform scales. a* expresses a uniform redness-greenness scale of color, while b* expresses a uniform yellowness-blueness scale of color. CIE L*a*b* is a popular color space choice for industries involved with reflective and transmissive products because it was derived from data obtained using reflective color samples.

Another uniform color space, CIE L*C*H*, is simply a coordinate transformation from CIE L*a*b* space to a polar coordinate system. The significance of L*C*H* lies in the plotting of its values. A color's chroma (saturation), plotted as the C* primary, is represented as the radial distance from the vertical L* (lightness) axis. The color's hue angle, H*, is plotted as the circular polar angle [3]. This representation allows for colors of a constant hue to be easily determined from hue angle while chroma within the hue is defined as the distance from the axis origin. This color space provides for analysis of colors in a manner similar to the system devised by Albert Munsell, a system focused on the relationship between colors. The reader is referred to the reference literature, specifically Hunt [7], for more information on the Munsell color system.

The idea behind uniform color spaces is that they have provisions for calculating a number representing how perceptually close two colors are to each other. This color difference value and the space dependent color difference equation used to compute it are the factors that make these color spaces practical and worthwhile. Simply put, the calculation of the color difference value amounts to determining the location of the two colors in the color space and calculating the difference between them [7]. The smaller the color difference value, the more similar the two colors.

Color Space Transformations

Transformations can be done to convert colors from one color space to another. These transformations are implemented as mathematical functions which convert the primaries of one color space to the primaries of another color space. The specifications for these transformations are part of the CIE's definition of the color space. The transformations can all be related back to CIE XYZ space, at least in the case of the color spaces presented in this thesis. It is also possible to convert, or reduce, a set of spectral data measurements to representative tristimulus values. This is done through the integration, on a wavelength by wavelength basis, of the spectral reflectance or transmission data in combination with the spectral information for the illuminant and the values given by Standard Observer color matching functions. Examples of the integration functions were defined previously for the SML and XYZ color spaces.

While an understanding of the mathematical processes that are required for color space transformations to be performed is important, and a detailed definition of them is possible, these equations will not be repeated here. Fortunately, computer software exists that can calculate the transformations between all relevant color spaces, as well as provide functionality for other color related operations.

Oregon Color Software

The Oregon Color Software is computer software that provides functionality for manipulating and calculating all varieties of color related data. It is a set of computer library functions, implemented as an application programmer interface in the C programming language, that can be used to perform color operations in any software program. The Oregon Color Software provides functionality to do device independent color calculations in numerous color spaces, fundamental colorimetric data about the CIE XYZ color specification system, and transformation routines between the XYZ color space and several other color spaces [12].

CHAPTER III

SCIENTIFIC VISUALIZATION USING COMPUTER GRAPHICS

If the often stated phrase "a picture is worth a thousand words" is true, then nowhere is that more the case than with the examination of large quantities of data. It is generally easier to spot trends in data using pictures rather than text, especially when the data represents large amounts of information. To this end, computer graphics provides a dynamic and accurate means to generate pictures using data.

Imagine a data set that provides a detailed description of an object's physical design, for instance the fender of a car. This data set consists of hundreds of values defining points on the surface of the fender in some coordinate measurement space. The points define the geometry of the fender: its size, shape, and curves. Now, try to imagine the geometry of that fender by simply examining the text of numbers that make the coordinate points. This task is quite difficult, if not impossible. Clearly it would be much easier to see the geometry of the fender by viewing a picture that is generated from the data points. This is a simple example showing what, at its most basic level, scientific visualization attempts to accomplish. Scientific visualization is a means for simplifying data analysis and understanding. In this problem, computer graphics is a tool that can be used to help achieve a solution.

Developments in computer graphics over the last several decades have allowed for significant advancements in the generation of images from data. These developments include the ability to display and manipulate data in three dimensions rather than two. This advantage is obvious because it allows for three dimensional objects to be modeled in a three dimensional view. The capacity to manipulate three dimensional objects, for instance by rotating them, is certainly more intuitive in a three dimensional space than in two dimensions. Also, the ability for a computer to render realistic, accurately colored images provides additional means for data analysis. The use of color in data visualization can be thought of as an added dimension in data representation. It allows for more advanced visualization techniques. Color can enable slight changes in a complex data set to be visualized [6].

In agreement with these descriptions is the definition of visualization in scientific computing assigned by a panel of computer graphics researchers which convened to discuss the topic:

> Visualization transforms the symbolic into the geometric, enabling researchers to observe their simulations and computations. Visualization offers a method for seeing the unseen. It enriches the process of scientific discovery and fosters profound and unexpected insights. In many fields it is already revolutionizing the way scientists do science.
>
> Visualization embraces both image understanding and image synthesis. That is, visualization is a tool both for interpreting image data fed into a computer, and for generating images from complex multidimensional data sets. It studies those mechanisms in humans and computers which allow them in concert to perceive, use and communicate visual information.
>
> - B. H. McCormick, et. al., quoted in
> *Advanced Animation and Rendering Techniques (p. 297).*

Scientific visualization has application in numerous, diverse areas. Many different disciplines, scientific or otherwise, have measurements that produce three dimensional data sets. Effective visual representation of the data creates a new understanding of it. Scientific visualizations are visual representations of numerical simulations. Using visualizations, doctors can plan surgeries, geophysicists can study the earth, and molecular modelers can produce new drugs [16]. Additionally, scientific visualization is frequently used in domains such as quantum mechanics, oceanography, and planetary science [15]. The areas of application of scientific visualization are increasing dramatically as advances in the field are made.

Visualization Model

Scientific visualization focuses on the analysis phase of a numerical simulation. Regardless of the type of simulation, it is during the analysis phase where computer visualization plays the largest role. The analysis of data is accomplished through operations that are common to all simulations. The steps in analysis done with a visualization system are filtering, mapping, and rendering. The process of analyzing data is a cycle that is executed repeatedly until all questions about the data have been answered [17]. Analysis begins after the data for a visualization has been input or generated.

Filtering involves converting the data of the simulation into another form. The new form is intended to be more informative. Examples of filtering operations include computing gradients from scalar fields, deriving flow lines from a velocity field, and extracting subsets of a data set [17]. Several of the visualization techniques described in the last section of this chapter are implemented as filtering operations.

In the next step, the filtered data is mapped to geometric primitives. These primitives are used to generate the computer graphics of display operations. They consist of entities like points, lines, spheres, surfaces, and polygons. A set of data can be mapped in many ways, with each mapping representing the data differently.

Lastly, rendering transforms the geometric primitives into images. Rendering parameters can be set to control the means by which the data is represented. Characteristics such as coloring, placement, illumination, and surface properties can be manipulated. The rendering step of analysis controls how the data will appear in the final display.

These steps form the cycle that defines the visualization model. This model is used as a basis for the design of many commercial visualization systems [17]. It provides a framework for all visualization applications that can be developed with a system.

Data Flow Architecture

Current, commercially available visualization packages generally incorporate modular, data flow architectures. A data flow architecture is a system design that allows for the flow of information (data) within the system to be controlled interactively by the system's user. Interactive control is established through the use of application components known as modules. These modules can be connected within the context of the visualization system's architecture to construct visualization networks [14]. A visualization network, in its usual form, will consist of modules that perform the following steps: reading or generating data, processing the data, applying visualization techniques to the data, and finally viewing the results. In this context, the modules that make up the network accept data as input, or generate the data, process it under the control of parameters that are interactively specified, and produce output data which is passed down the network to the next module, or viewed in its final form [14]. Many of the modules correspond to functions of the cycle that define the visualization model.

Application development is done in a visual environment. The application components in the system appear as some form of visual icon which are defined, assembled, and manipulated through common computer interaction techniques such as mouse and keyboard operations. This development approach is object-oriented. In many visualization programs, support is provided for such object-oriented techniques as data and method encapsulation, class inheritance, templates and instances, object hierarchies, and polymorphism.

Visualization Techniques

Information with spatial components, collected as a volume of data, is typically represented as a set of discrete points in three dimensional space. A relevant example of such a volume of data is the color gamut of a color reproduction device. Three dimensional space also can be used to represent two dimensional data that has associated time components, for instance as a series of stacks of the two dimensional data, with each stack representing one time period [18]. The representation of these types of data in a three dimensional volume is referred to as volumetric rendering. In order to extract useful information from the volume of information a large number of data visualization techniques have been devised to allow the data to be viewed in meaningful ways. Many of these techniques allow for data that is internal to the structure of the volume to be examined. Several of these techniques will now be addressed.

Data Slices and Cuts

Volumetric data can potentially reveal more of its structure when it is dissected into slices or cut into sections. This is especially true when the data is viewed as a solid. The ability to take a data slice, i.e. a two dimensional cross section, through any plane of the volume can easily reveal detail that is impossible to see otherwise. Similarly, the capacity to cut away sections of a volume above or below some threshold can offer insight into the interior of the volume. A simple example would be taking slices of data from something like a medical magnetic resonance imaging scan to reveal portions of the data from the internal organs of the body. Slicing and cutting volumetric data is among the most commonly used visualization techniques.

Data Exteriors and Edges

A finite set of data will normally exist in its own data space, with unique boundaries and dimensions. Useful information can be extracted from the data simply by identifying the data boundaries and data space extents that are represented [2]. All data sets exhibit two containment properties.

One containment property is the boundaries that define the region into which all the data values are held. This data boundary is the data set that inclusively contains all the data points. The maximum and minimum values of the data in each of the dimensions being considered make up these data boundaries. When viewed, these boundaries can take the form of visible edges, surfaces, or other entities [2].

The second property, the data space extents, defines the limits imposed on the data set by the coordinate system in which it resides. In most cases, the data space extents form the metaset for the data boundaries, or the maximum and minimum values of the coordinate system containing the data [2]. In viewable form, data space extents are also represented as edges or surfaces.

Displaying the exteriors of color gamuts is a useful way to interpret their discrete data values. Modules that determine and display boundaries and extents are common in visualization programs.

Isosurfaces

An isosurface is the three dimensional surface representing the locations of a constant scalar value within a volume [6]. Scalar data stored at each point in a volume might represent some variable value like temperature in atmospheric data, or density in a cloud of particle data. Each point in the isosurface holds the same constant scalar value. When rendered, an isosurface is a visible contour defining a surface. Isosurfaces are another way to render volumetric data. Many techniques exist for generating isosurfaces including the Marching Cubes and Dividing Cubes algorithms [6].

Isosurfaces are used to give clear representations of a critical value in a data set. Operations for working with isosurfaces include the ability to dynamically set the scalar value of interest. The colors used in representing different values of the scalar variable can also be set interactively. With these techniques, isosurfaces can be used to show concentrations of values in a volume. By interactively changing the value, behaviors in data can also be identified with isosurfaces.

Data Glyphs

The ability to use three dimensional graphic icons to represent values in a data set, or place markers at data values in three dimensional space, is a powerful visualization technique. Graphic objects used to represent data values at a single location in space are called glyphs. Glyphs are useful for allowing large volumes of information to be interpreted in one view because they can represent different variables in a data set. Arrow glyphs, called vectors, can represent the movement of values in a set of data, or between multiple sets of data. Glyphs are also used as intelligent markers. An intelligent marker, as an example, conveys information through not only its location and shape, but also its size and color. Used in this way glyphs can offer information that is otherwise not easily interpreted from a visualization.

Data Interaction and Probing

Interacting with data can provide a visualization program user with intuition about a data set that may not otherwise be possible. Operations that can be performed in realtime, such as rotating and scaling data volumes, or myriad other operations, allow for views of data sets from any angle and perspective. Data that is invisible from one view can suddenly reveal itself when rotated to a new position.

The capacity to probe information from the contents of a viewed volume of data allows for mental connections to be made between actual data values and their location in three dimensional space. Determining position and value relative to other points in a data set can help make comparisons between them. Probing can also be used to generate events within a visualization application. Probing a point in a data set can cause its value to be sent to other processing modules that might use the value in performing some calculation.

Appearance Manipulation

Often more information can be gleaned from a data set by manipulating the properties used to control its appearance. Operations for appearance manipulation can vary from changing the synthetic lighting conditions used to illuminate a volume of data, altering its color, or modifying surface properties such as transparency. If only the shape of a volume is the concern of a visualization, its properties could be changed so that it is displayed monochrome, potentially eliminating any influence of color during viewing.

Appearance manipulation techniques can prove especially useful in the case of viewing multiple volumes. If the volumes intersect or overlap, modifications can be done to one or all to make them more easily identifiable. For example, if one volume encompasses another its surface could be rendered transparent to allow the internal data set to be seen.

Combinations

Almost all of the described visualization techniques can be used in combination with one another. Applied in sequence, a visualization technique can transform the data that is generated by a previous one. Examples include performing a slice on an isosurface and cutting values from a volume represented as glyphs. The flexibility and power that permit the simultaneous use of multiple visualization techniques are among the greatest strengths of visualization programs.

CHAPTER IV

COLOR GAMUT VISUALIZATION SYSTEM

A computer software tool has been developed that can be used by engineers to analyze the color gamuts of color reproduction devices. The color gamut visualization system was created with a commercially available visualization package: Advanced Visual Systems Inc.'s Express Developer's Edition (AVS Express). Through features of AVS Express and the design of the gamut visualization system, a flexible framework exists to provide solutions to most any gamut visualization problem.

AVS Express

AVS Express is an object-oriented, visual development tool that provides a software engineer with facilities for building sophisticated scientific visualization applications. The generation of visualization applications is accomplished by creating reusable software components known as modules. Modules are combined to create complete applications.

Express provides many predefined modules that can be used in visualizing data. These modules support operations for analyzing, manipulating, displaying, and interacting with data. Custom modules are created using a combination of existing building-block components and software code generated by the developer. Express offers support for both the C and C++ programming languages through a set of application programmer interface (API) library routines.

Visual development is done in AVS Express through the Network Editor. Within the Network Editor, modules appear as rectangular shaped icons. These icons are manipulated by the developer with mouse-driven operations. Modules are defined and assembled in the application workspace of the Network Editor. The application workspace is also where modules are connected together to construct complete applications. Module connection is achieved by creating connection lines between input and output ports that appear on the modules. Input and output ports represent the entry and exit locations of data for a module. These ports appear as colored blocks on the edges of the module icons.

An important feature of the Network Editor is libraries. Libraries provide means for storing and reusing modules. Express supports a hierarchical organization of libraries. Modules that are created by a developer can be stored as objects within the libraries. Stored modules are used in an application by selecting the module and "dragging and dropping" an instance of it into the application workspace of the Network Editor. The operation of instantiating objects with a mouse action is one of many object-oriented features in Express.

AVS Express organizes its predefined modules into several collections called kits. The data visualization kit contains the modules for reading, manipulating, transforming, and analyzing data sets. The user interface kit holds modules that build platform-independent graphical user interfaces for visualization applications. The functionality in the user interface kit allows for standalone visualization applications to be constructed. Other kits with uses unrelated to gamut visualization are also provided.

AVS Express was chosen as the visualization development environment for the color gamut visualization system because it offers the most direct solution to design requirements and the highest graphics performance for the target hardware platform. An additional factor is that it runs on many common computer hardware platforms, thereby making the color gamut visualization program available to a large audience of users.

System Design and Components

The architecture of the color gamut visualization system consists of three basic types of modules. Source modules generate data by executing an algorithm or by reading a data file. Transformation modules filter the data or change it into another form. Lastly, terminal modules display the data or write it to a file. The system contains several modules of each type. Modules are stored in the system in one or more libraries representing each type. These modules form the base that an analyst can use and build upon to solve his or her own gamut mapping or gamut visualization problem [14].

Source Modules

Source modules fall into the categories of generators and readers. Both categories share common properties. Source modules have no input ports and one output port. The output of all source modules is a data stream of tristimulus values which is passed into a visualization network from the output port. Encoded within the output data stream are the color space identification and parameters relevant to a particular color space.

Generator Source Modules

A generator source module that produces numeric data representing discrete tristimulus values is provided in the system. The output data of this module is RGB tristimulus values. The density of the data points within RGB space from this generator can be controlled interactively with standard, graphical user interface components. These controls set the number of numerical divisions which the module's algorithm uses in tristimulus production.

Reader Source Modules

Two different reader source modules, which read data from input files, have been developed. Each module incorporates interface components for file selection.

The tristimulus data points reader source module inputs tristimulus values in any standard color space and directs them to the output port. The module identifies the color space represented in the file and encodes that information in the output data stream. Additional relevant information contained in the file can also be encoded. This information may include data such as an identifier for the illuminating light source or the illuminant's white point specification.

The spectral data reader source module reads color sample information from a file. The data is in the form of wavelength and percent reflectance pairs. The module supports a varying number of samples per file and a varying number of wavelength measurements per sample. The spectral information for each sample is integrated to reduce it to its corresponding tristimulus values. Integration is done using one of the standard CIE light sources as the illuminant. The output tristimulus values are in the CIE XYZ color space.

Transformation Modules

A data stream of tristimulus values produced by a source module can be passed into a transformation module to be filtered or transformed. Transformation modules have one or more input ports and one or more output ports. The transformation modules of the color gamut visualization system form the largest collection within the system. Transformation modules are categorized as color space transformation modules, color space identification modules, AVS mesh builder modules, and visualization operation modules. Libraries maintaining modules in each category are found in the system.

Color Space Transformation Modules

Tristimulus values in the color gamut visualization system can be expressed in standard color spaces such as CIE XYZ, xyY, L*a*b*, L*u*v*, and L*C*H*. Conversion between these spaces is done by connecting the input port of a color space transformation module to the output port of a source module or another color space transformation module. An analyst can display tristimulus values for a gamut in the color space of interest by using these modules.

All color space transformation modules have one input port and one output port. The data type for both ports is a stream of tristimulus values. Conversion processing between color spaces is done using functions provided in the Oregon Color Software accessed through the C API of Express. Use of the Oregon Color Software promotes accuracy in color space conversions, and supports extensions for future color spaces.

The CIE XYZ color space is the foundation of the CIE color specification system. Because of this, conversions between color spaces in the color gamut visualization system generally involve modules that transform to or from CIE XYZ space. An example is the conversion from RGB values to the L*a*b* color space. An intermediate step in this process is the conversion from RGB to XYZ. The output of the module for this conversion is directed to the module for XYZ to L*a*b* transformation. The conversion of RGB values to CIE XYZ space requires additional information. This information includes characteristics of the original primaries of the monitor. The module for this transformation presents a graphical user interface for setting the chromaticity coordinates of the monitor's red, green, and blue phosphors and the chromaticity coordinates and luminance for the white point. These values determine what colors are shown when the phosphors are combined, thereby determining the location and size of the monitor's gamut in XYZ space [13]. This data is encoded with the tristimulus data stream that the module produces and can be used in later color space transformations.

Color Space Identification Modules

A single color space identification module is provided in the system. Its purpose is to produce data that can be used by terminal modules in the process of displaying gamuts. The data that is produced by the module specifies the color space of the gamut that will be displayed and also provides normalization information to the terminal modules. This data gives terminal modules the necessary information to label the coordinate axes that are displayed. It also identifies the extents of the coordinate space in which the gamuts will be rendered. The color space identification module has a single input port which accepts a data stream of tristimulus values. The single output port produces a data type containing the identification and normalization information.

Mesh Builder Modules

The AVS Express mesh data type is a compact encoding of three dimensional data. It is used by the Express program for both visualization operations and conversion to a final, displayable data format. Mesh builder modules transform an incoming tristimulus data stream to the mesh data type. This mesh data type can be passed to visualization modules, or directed to a terminal module for display. The mesh builder modules of the gamut visualization system have one or more input ports which accept tristimulus values. Each mesh builder module has two output ports. One port produces the mesh data type for use by visualization modules further in the network. The second output port produces the displayable format data stream, which can be sent directly to a terminal module if no further transformations are required. Three mesh builder modules are found in the gamut visualization system libraries.

The color surface mesh builder creates a mesh representing a gamut as a three dimensional solid volume. The output of this builder is used to view a gamut as a solid object. Internal components of the module accept tristimulus values in any color space and convert them to the mesh format. A step in the conversion involves determining the RGB information for each tristimulus data point so that an appropriate color value can be stored in the mesh with the point's coordinate information. This step provides the color values for the rendered surface. The color surface mesh builder has a single, tristimulus data stream input port and the two output ports described previously.

The glyph mesh builder creates a mesh representing a gamut as a collection of individual points in three dimensional space. The glyph mesh builder incorporates a user interface component to allow the type of marker that is displayed at each point to be chosen interactively. The different types of supported markers include points, arrows, crosses, diamonds, and other geometries. The input and output ports of the glyph builder are identical to those of the color surface mesh builder.

The vector difference mesh builder is unique because it has two input ports for tristimulus data streams. The vector difference mesh builder is used to show the differences between two sets of discrete color data. The mesh that is produced by this module represents the change in colors between data sets with directed vectors. The head of each vector represents a point in one data set, while the tail represents the corresponding point in the other. The inputs to the vector difference mesh builder can be generated by any of the source modules, providing that both input data sets have the same number of tristimulus values. Output ports of the vector difference mesh builder are identical to the other mesh builders.

Visualization Modules

Visualization modules provide the functionality for utilizing scientific visualization techniques in the analysis of color gamuts. The mesh data output of a mesh builder module can be connected to the input port of a visualization module for data transformation or manipulation. Visualization modules have a single input port which accepts a mesh format data stream and one or two output ports. All modules in this category have an output port for passing the displayable data format to terminal modules. If appropriate to the visualization technique the module implements, it may also have a second port which outputs a mesh data stream. This mesh data stream, representing the manipulated data, can be sent to other visualization modules. In this way, multiple visualization modules can be used sequentially.

The edge detection visualization module can be used to detect and display the boundaries of a gamut. This module is useful when combined with the glyph mesh builder module. Used in combination, the boundaries of a gamut can be made visible, while still allowing for individual tristimulus data points to be seen.

The slice plane visualization module will perform a slicing operation through a color surface gamut mesh. The slice appears as a two dimensional, color cross section of the gamut volume. User interface controls facilitate the control and manipulation of the location and orientation of the slice plane.

The cut plane visualization module is used to cut away portions of a gamut volume allowing for interior sections to be viewed, while leaving the remainder of the gamut intact. The cut plane module operates similarly to the slice plane module with controls to manipulate the cutting plane's location and orientation.

The library of visualization modules in the color gamut visualization system is the most dynamic. As new visualization techniques are developed for interpreting gamut data, modules that support the techniques are added to this library.

Terminal Modules

The final, display format data stream of a mesh builder module or a visualization module is directed to a terminal module. Terminal modules determine how a gamut appears when it is finally displayed by AVS Express. The terminal module also allows a user to interact with the gamut once it has been rendered [14].

The color gamut display module has two input ports and no output ports. One input port accepts display format data from a mesh builder module or a visualization module. This port allows for multiple input connections to be made to it, providing for the display of multiple gamuts. The second input port accepts the data stream containing the color space identification and normalization information. This data is the output of the color space identification module.

Many of the most powerful features of the visualization program are incorporated into the color gamut display module. Several visualization operations that are common to any gamut display application are internal to the module.

Interactive gamut control functionality is provided in the color gamut display module. Interaction with the displayed data is allowed in several ways. This interaction includes the ability to exhibit the numeric values for a tristimulus data point through probing operations. It also includes means for manipulating gamuts in three dimensional space.

Operations for gamut appearance manipulation are implemented in the color gamut display module. Functionality for changing gamut environmental factors, such as lighting conditions and background colors, are provided. Surface properties of a gamut, such as transparency, can be controlled. Transparency is useful when displaying multiple gamuts. It allows a gamut that obscures another to be rendered transparent to permit the obscured gamut to be seen. Gamuts can be made to appear monochrome through controls of the color gamut display module. This is useful when only the size and shape of a gamut is of concern, or in multiple gamut analyses.

The color gamut display module also controls secondary output of displayed data. Any image that can be rendered using the color gamut display can be saved to a file in a variety of image formats. Additionally, the color gamut display module includes PostScript printing controls to allow for hardcopy output.

System Use and Functionality

The minimal color gamut visualization application consists of four modules: a source module, a color space identification module, a mesh builder module, and a color gamut display module. The visualization network created by these modules displays a single color gamut. More complex visualizations can be created by adding color space transformations modules and visualization modules.

Creating Gamut Visualization Applications

To create the minimal application, the first step is instantiating the modules. Each of the four modules is first located in the color gamut visualization system libraries. Once selected with a mouse click, it is “dragged and dropped” into the application workspace of the Express Network Editor. Next, the appropriate input and output ports of the modules are connected. Modules are connected by initially selecting the output port of one module with a mouse click operation. The connection is completed by highlighting the desired connection line. Express automatically draws connection lines from the initially selected output port to all potential input ports of all modules in the Network Editor. This feature makes it simple to identify valid connection ports.

Figure 7 shows one possible version of a minimal color gamut visualization application. It consists of a tristimulus values reader module, a color space identification module, a color surface mesh builder module, and a color gamut display module. The result of creating this minimal application is visible in Figure 8. This volume represents the gamut of a color printer in the L*a*b* color space.

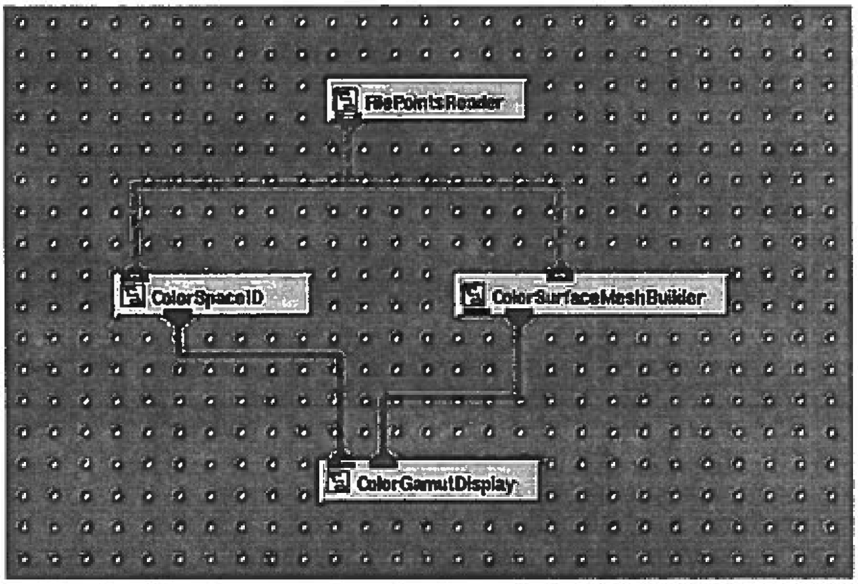

Figure 7. A minimal gamut visualization network which reads tristimulus values from an input file and displays a colored surface gamut.

While it is the most basic, this application displays critical features of the color gamut visualization system. It demonstrates the means for connecting modules and represents all steps in the flow of data within a gamut application: data generation, transformation, and display. The steps in the flow of data through a gamut visualization application are designed to create a simple mental model for a user. The user of the system can maintain this mental model as a basis for all gamut visualization applications that are created.

Key features of the gamut visualization system promote this mental model. The categorization of modules into source, transformation, and terminal modules, and their division into libraries within the color gamut visualization system, make module location

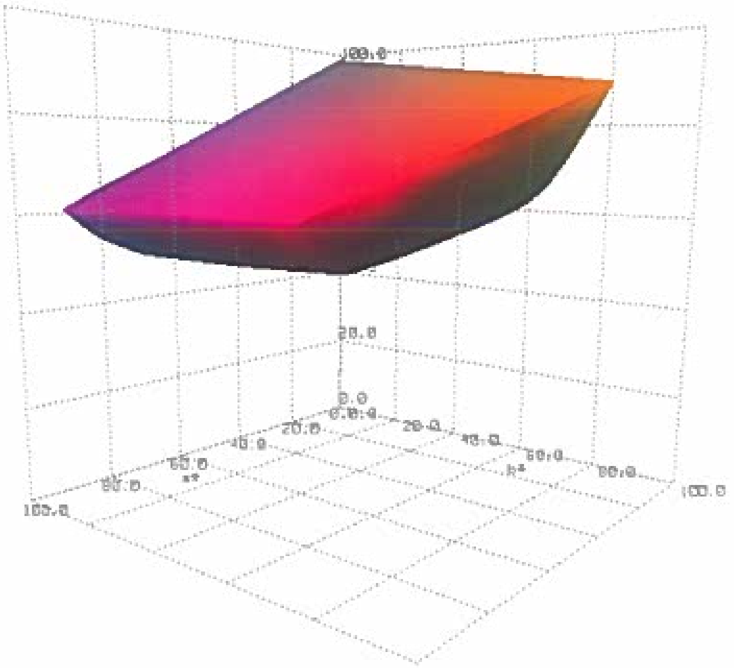

Figure 8. A color DeskJet printer gamut displayed in the L*a*b* color space.

and identification straightforward. Also, the operation of connecting modules is simplified in several ways. As mentioned, when a module's port is selected for connection, AVS Express automatically draws all possible, valid connection lines to ports on all modules currently in the Network Editor's application workspace. Additionally, the ports on modules within the color gamut visualization system have been color-coded to identify the type of connection they support. This allows for immediate identification of the appropriate location of a module in an application. Module ports which accept tristimulus data streams are identified with hatched red, green, and blue colors. Mesh data ports have hatched blue and black coloring. Ports which carry color identification and normalization data have magenta and blue hatched coloring. Input and output ports which accept displayable data are pure red. Port color-coding is visible in Figure 7. Lastly, the gamut visualization system will not allow connections to be made between ports of incompatible types. Connection lines will never be displayed by Express between ports that are incompatible.

Interacting with Displayed Data

Operations for interacting with displayed data in the gamut visualization system are intuitive. Gamuts are manipulated directly in the viewing window through mouse operations. Support is included for standard manipulation functions, such as rotation, translation, and scaling, as well as more specific functions, such as object picking and probing.

Rotating, scaling, and transforming gamuts are all done with similar mouse operations. Each is accomplished by clicking the displayed object of interest with the mouse cursor and dragging it in the direction in which change is desired. Rotation is the default operation of mouse interaction in the Express data viewer window. The other two types of standard manipulation are chosen by selecting a toggle button to indicate which function is desired. An example manipulation would be to rotate a gamut around its horizontal axis. The gamut is first selected with a mouse click on its surface and then dragged in an upward or downward direction. Speed of rotation is set relative to the speed of the mouse drag movement. Scale and transformation operations behave similarly.

The interactive techniques of picking and probing are done with a combination of mouse and keyboard controls. Picking allows for individual display elements to be selected. It is a necessary operation for performing functions such as appearance manipulation. For example, to make a gamut transparent, it must first be selected through picking. Once the gamut has been picked, the object editor in Express can be used to set appropriate surface properties. Probing allows a user to display the numeric values for a tristimulus data point by selecting an area of the displayed data with a mouse click. Figure 9 represents a gamut in CIE L*a*b* space with probed tristimulus values displayed.

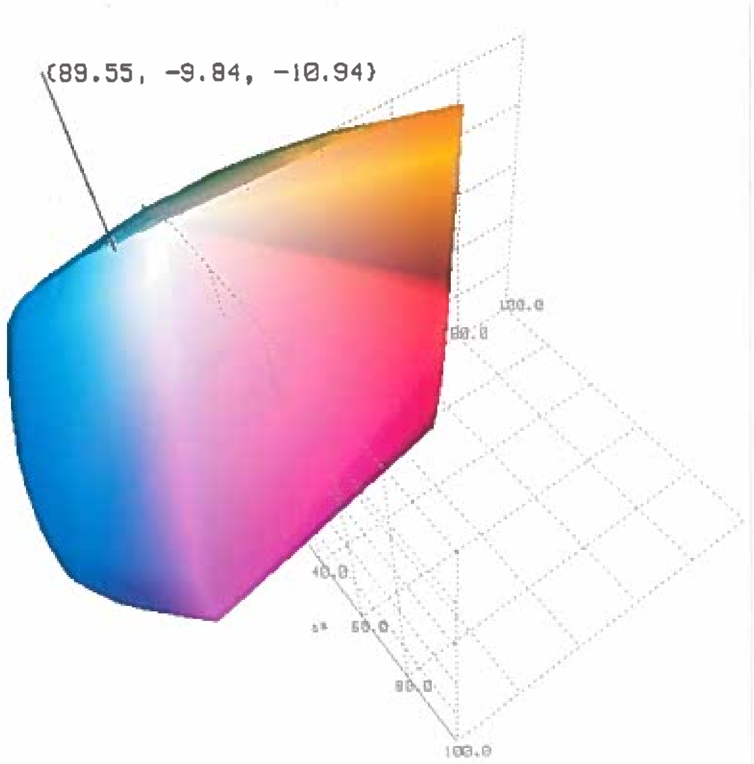

Figure 9. A color DeskJet printer gamut with probed value displayed.

Creating Advanced Gamut Visualizations

A main motivation in the design and creation of the color gamut visualization system was to provide means for creating advanced gamut visualizations. Advanced visualizations may involve analyzing the display of multiple gamuts, or the display of the movement of colors between two sets measurements. Means for analysis of color data in this way was previously unavailable to engineers. It is a strength of the gamut visualization program.

Multiple Gamut Display

Multiple gamuts can be displayed simultaneously with the system. Gamuts from different reproduction devices can be compared, or gamuts from the same device under different conditions can be compared. By observing the location of the surface of one gamut relative to that of another, an engineer can identify colors from one device that cannot be reproduced by the second. Figure 10 shows the simultaneous display of the gamut of a color monitor and the gamut of a color printer in CIE L*a*b* space. The monitor gamut is rendered transparent to reveal the solid printer gamut it encompasses. The module network used to create this visualization is seen in Figure 11. To further aid in the analysis, a visualization module, such as the cut plane module, could be used. The cut plane removes the hull of the outer gamut to reveal the inner one. This is visible in Figure 12, which is created with the modules in the network of Figure 13. An alternative visualization technique would be to display one or both gamuts using the glyph mesh

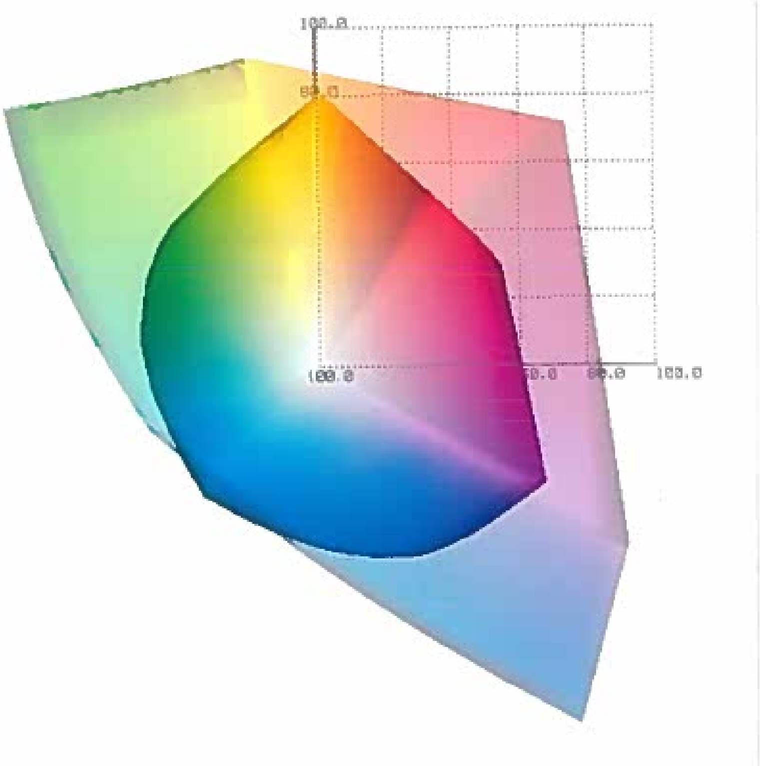

Figure 10. A monitor gamut (transparent) encompassing a color DeskJet printer gamut (solid) in L*a*b* space.

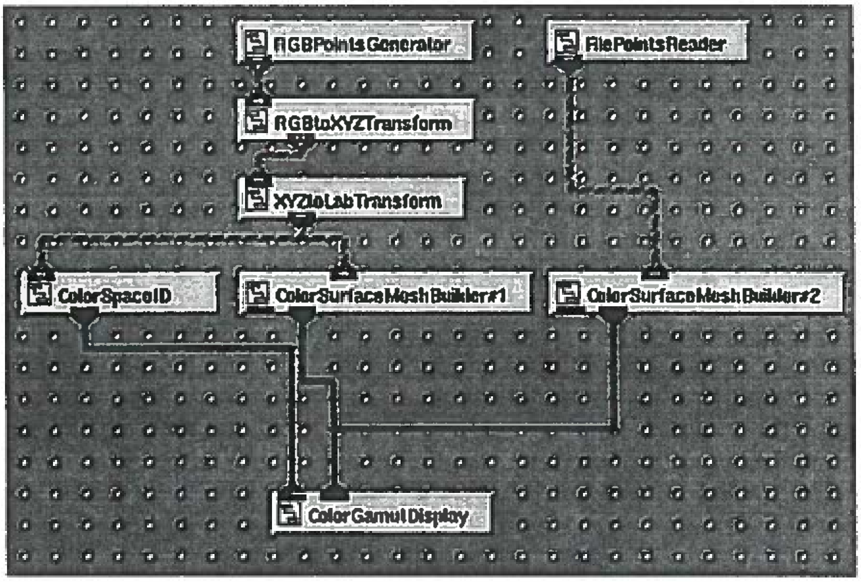

Figure 11. The module network used to construct the multiple gamuts displayed in Figure 10.

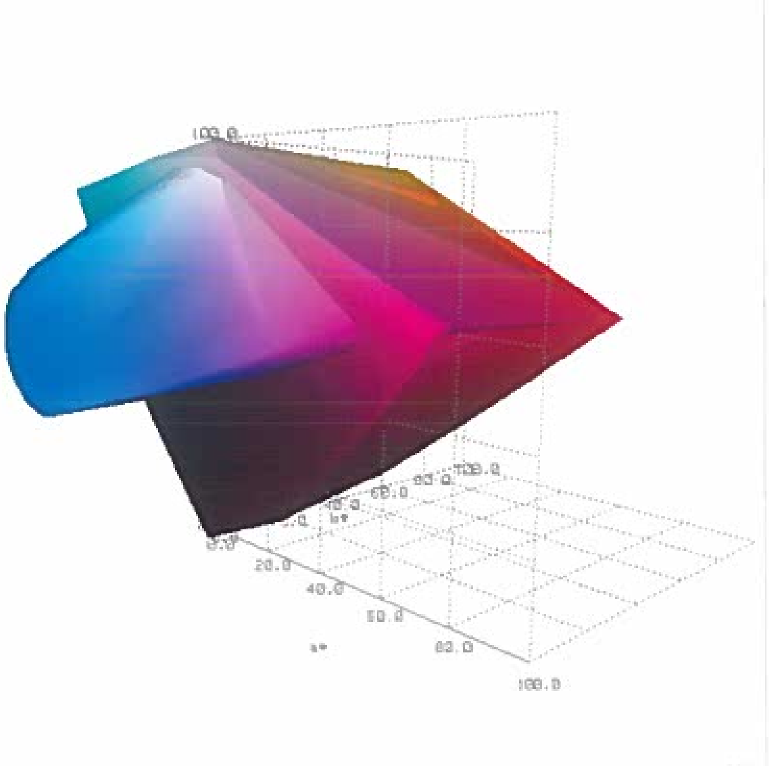

Figure 12. A monitor gamut, which has been partially cut away using the cut plane module, encompassing a color DeskJet printer gamut in L*a*b* space.

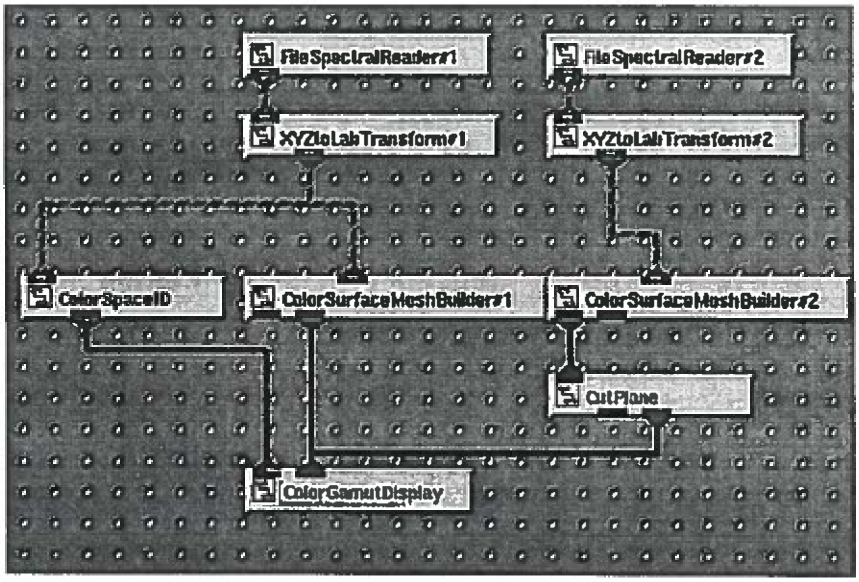

Figure 13. The gamut visualization application used to create the visualization of Figure 12.

builder module. Each gamut could be represented with a different glyph object, for instance one as diamonds and the other as points. A single gamut represented with glyphs is shown in Figure 14. No bound exists on the number of gamuts that can be displayed at one time. Appearance manipulation techniques, combined with the use of surface mesh and glyph mesh builders, provide myriad ways to visualize multiple gamuts.

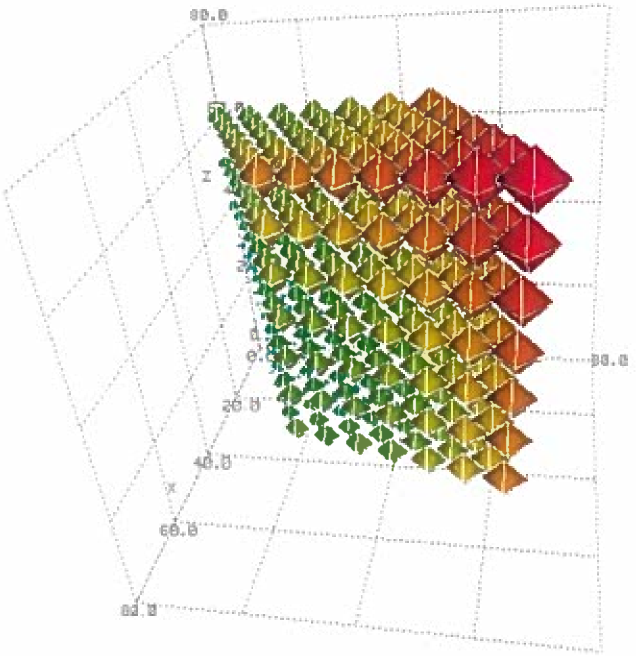

Figure 14. A monitor gamut in the CIE XYZ color space with tristimulus values shown as diamond glyphs.

Gamut Difference Display

The ability to display the difference between two gamuts through vectors is a needed tool. Vectors represent the movement of a data point between two locations.

With color data, they can show the change between colors. A color reproduction engineer might need to determine the difference in the output of a color printer with different ink sets. Another situation may require analyzing data showing a printer's output on different paper types. Alternatively, the effects of aging on the inks used in a color printer may need to be determined. The vector difference mesh builder provides a solution to all these problems. Figure 15 and Figure 16 show a vector difference gamut visualization and the module network that created it, respectively. The visualization shows the difference between output on two paper types, plain and gloss, for a color DeskJet printer.

Used in combination with other mesh builder modules and visualization modules, examining differences between data sets can be an extremely powerful technique for gamut analysis. The solid volumes of two gamuts can be rendered with the difference between their corresponding data points displayed as vectors. In this way, an analyst can quickly see the size, shape, and extents of each gamut, as well as the exact path of color movement between them. A visualization of this type, and its construction, is shown in Figures 17 and 18.

The advanced visualization techniques presented to this point only scratch the surface of the potential uses of the gamut visualization system. They represent a set of common techniques for visualizing color data and are the basis of many possible analyses. One of the most important aspects of the system's design is that its functionality will evolve with its use. Rather than having a locked feature set, its possibilities for gamut visualization are potentially boundless.

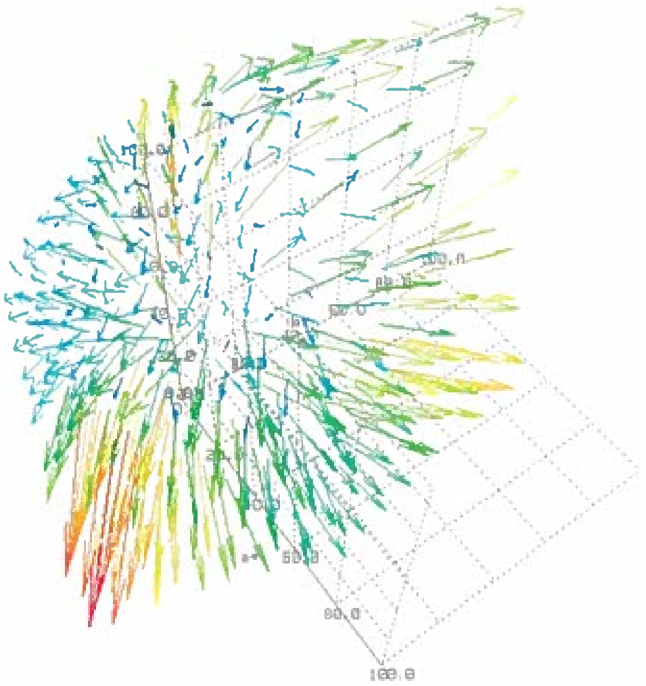

Figure 15. Output of the vector difference mesh builder used to display the difference in the gamuts produced by the same printer on plain (innermost values) and glossy paper (outermost values).

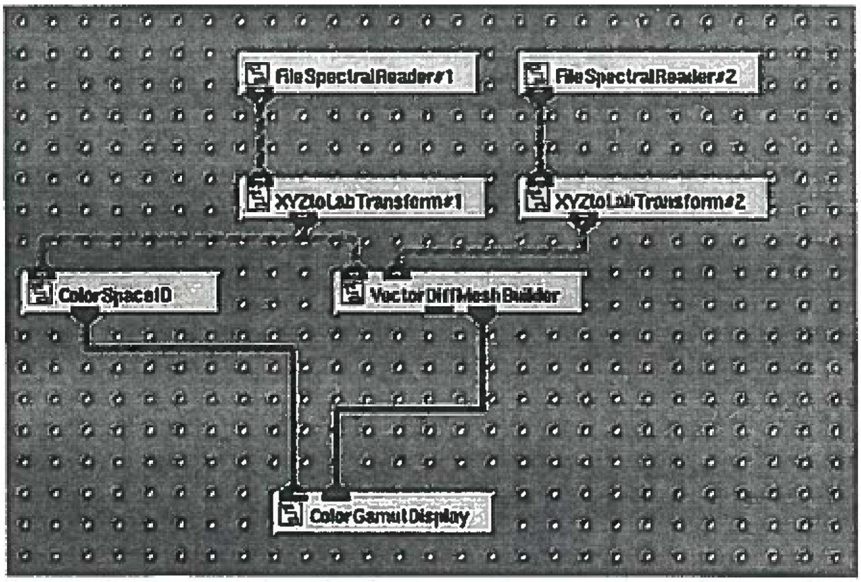

Figure 16. The gamut visualization network used to create Figure 15.

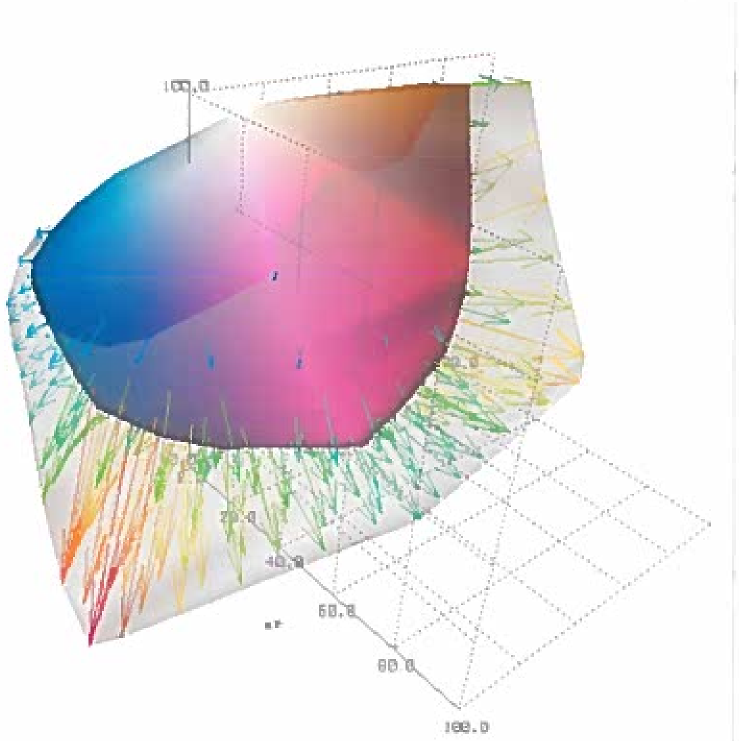

Figure 17. A visualization showing the difference between the output of a color DeskJet printer on plain versus glossy paper with vectors and surfaces.

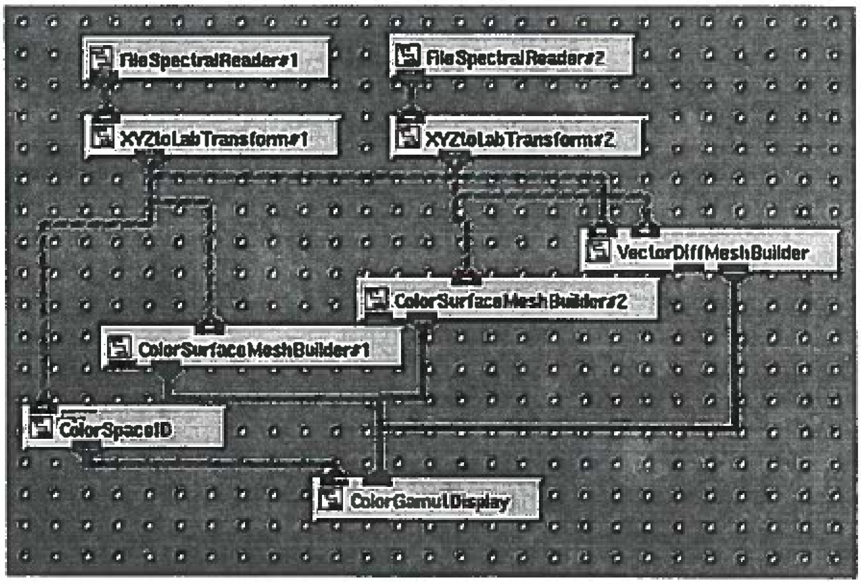

Figure 18. The color gamut visualization system module network used to create the visualization shown in Figure 17.

Color gamut visualization applications like those described in this chapter can be saved in AVS Express. This feature is beneficial with advanced networks, whose construction can be time consuming. An analyst can recall previously saved applications without having to rebuild a network from scratch. With this functionality, a set of common visualization applications can be created and repeatedly used.

CHAPTER V

CONCLUSIONS

A software tool was developed to help engineers evaluate the color gamuts of color reproduction devices. The color gamut visualization system is designed around a data flow architecture. Modules are the building blocks of the system and are connected together to construct application networks. The modules of the system are categorized as three types: source modules, transformation modules, and termination modules. Source modules generate or read data and direct it into the network. Transformation modules filter incoming data or change its form. Finally, termination modules display the data or produce secondary output. The color gamut visualization system allows advanced visualization applications to be constructed. These applications employ leading edge visualization techniques. The system can be extended through the creation of new modules.

The color gamut visualization system is a powerful, flexible tool that can be used by color analysts to aid them with gamut visualization problems. Its power results from the use of a commercial visualization program, which handles the complexities of accurately and rapidly displaying realistic computer graphics. The flexibility of the tool lies in the ease with which extensions and modifications can be made to the system. The system makes complex visualizations available which were previously not possible.

The development of the color gamut visualization system precludes the need for an analyst to develop and write a computer program from scratch. It also allows the analyst to work without restrictions imposed by a single monolithic color space visualization program. The gamut visualization program can be easily extended through the development of new modules. The new modules may support new color reproduction technologies, allow transformations for non-colorimetric data manipulations, or access new scientific visualization techniques. The modularity of the system facilitates its use between individuals, and could eventually form the basis for a standard way of communicating data between color analysts [14].

As a teaching aid, the color gamut visualization system offers a means of visual reinforcement which was previously unavailable to anyone involved with the study of color. An instructor can use the system, in conjunction with standard teaching techniques, to demonstrate principles of color science in a practical environment to speed learning. Students can gain hands on experience with the implementation and practical use of color theory.

Ultimately, the worth of any software system lies in the value it holds to its users, not its designers. The color gamut visualization system has the technical merit to make it valuable to a large and varied user base. Future additions to the system can extend its validity and usefulness, as well as its lifetime.

BIBLIOGRAPHY

[1] Advanced Visual Systems, Inc. *AVS/Express User's Guide, Release 3.0.* Waltham, MA, 1996.

[2] Advanced Visual Systems, Inc. *Visualizing Your Data with AVS/Express, Release 3.0.* Waltham, MA, 1996.

[3] Commission Internationale de l'Éclairage. *Colorimetry, Official Recommendations of the International Commission on Illumination.* Publication CIE No. 15, E-1.3.1, 1971.

[4] H. S. Fairman. *The Calculation of Weight Factors for Tristimulus Integration.* Color Research and Application, 10, 4, 1995, pp. 199-203.

[5] J. D. Foley, A. vanDam, S. K. Feiner, and J. F. Hughes. *Computer Graphics Principles and Practices,* Second Edition. Addison-Wesley, Reading, MA, 1991.

[6] R. S. Gallagher. *Computer Visualization: Graphics Techniques for Scientific and Engineering Analysis.* CRC Press, Boca Raton, FL, 1995.

[7] R. W. G. Hunt. *Measuring Colour, Second Edition.* Ellis Horwood Limited, Cheshire, West Sussex, England, 1991.

[8] Illuminating Engineering Society of North America. *Color and Illumination.* IESNA, New York, NY, 1992.

[9] D. B. Judd and G. Wyszecki. *Color in Business, Science, and Industry, Third Edition.* John Wiley and Sons, New York, NY, 1975.

[10] Light Source Computer Images, Inc. *Colortron User's Guide: Color and Measurement Primer, First Edition.* Lakespur, CA, 1994.

[11] G. W. Meyer. *Colorimetry and Computer Graphics.* Cornell University, Program of Computer Graphics Report, Number 83-1, 1983.

[12] G. W. Meyer and L. S. Peting. *Oregon Color Software Manual.* University of Oregon, 1993.

[13] G. W. Meyer, L. S. Peting, and F. Rakoczi. *A Color Gamut Visualization Tool.* Proceedings from IS&T/SID Color Imaging Conference: Transforms and Transportability of Color, 1993, pp. 197-201.

[14] G. W. Meyer and C. A. Robertson. *A Data Flow Approach to Color Gamut Visualization.* Submitted for publication, University of Oregon, 1997.

[15] L. Rosenblum, R. A. Earnshaw, J. Encarnacao, H. Hagen, A. Kaufman, S. Klimenko, G. Nielson, F. Post, and D. Thalmann. *Scientific Visualization: Advances and Challenges.* Academic Press, San Diego, CA, 1994.

[16] L. Rosenblum, G. Nielson. *Visualization Comes of Age.* IEEE Computer Graphics & Applications, 11, 3, 1991, pp. 15-17.

[17] C. Upson, T. Faulhaber, D. Kamins, D. Laidlaw, D. Schlegal, J. Vroom, R. Gurwitz, A. vanDam. *The Application Visualization System: A Computational Environment for Scientific Visualization.* IEEE Computer Graphics & Applications, 9, 4, 1989, pp.30-42.

[18] A. Watt and M. Watt. *Advanced Animation and Rendering Techniques: Theory and Practice.* ACM Press, New York, NY, 1996.