David H. Brainard*
Brian A. Wandell

Department of Psychology
Stanford University
Stanford, CA 94305

Calibrated Processing of Image Color

We describe the design philosophy and image representation of a calibrated color image processing system. The design emphasizes the representation of the spectral functions necessary for device-independent acquisition and display of color data. These basic design principles can be incorporated into a new generation of image processing software that integrates both spatial and spectral image processing.

Introduction

Monochrome image processing packages represent image data with a single number for each image location. The stored value can represent a number of distinct types of image data. The number may represent the intensity of light emitted by a monitor or the response of a camera to incident light. It is rarely necessary to include additional information about monitor or camera spectral properties in the image representation. For example, when we render a monochrome image on a video display the spectral power distribution of the phosphor is of little interest since it cannot be controlled. When the same image data are displayed on two monitors with different phosphors, the images will look different.

Color image data are fundamentally different. When the same image data are displayed on two monitors with different sets of phosphors, it is possible to use display procedures that cause the rendered images to appear identical to a human observer.† To build such display procedures, the phosphor spectral power distributions must be explicitly available to the image processing software.

In the impoverished world of monochrome imaging, the data offer almost no hope of separating the effects of the various physical factors involved in natural image formulation: illuminant spectral power distributions, surface reflectance functions, and camera sensor responsivities. Using calibrated color image processing methods,[3–6] it is possible to estimate separately the influence of these physical factors upon the image data. To use calibrated color image processing methods, the camera sensor responsivities must be known and the image data structures must permit specification of illuminant spectral power distributions and surface reflectance functions.

For example, consider the problem of creating a digital archive of a painting. If the archive includes only camera sensor responses, then the representation confounds the effects of the illuminant spectral power distribution, the painting's surface reflectance functions, and the spectral responsivities of the camera sensors. Of these physical factors only the surface reflectances are intrinsic to the painting. Preferable to storing the camera sensor responses is to store estimates of the painting surface reflectances. This separates the archival image data both from the particular illuminant under which it was acquired and from the properties of the image acquisition hardware. A surface representation would allow future generations to view the painting rendered under many different possible conditions.

In this article, we describe the design principles underlying a software package we have written to perform calibrated color image processing. We discuss the representation of basic color image data using two file types, image files and basis files. We illustrate the power of the representation by showing how it facilitates device independent display of camera data. Then we discuss the representation of spectral functions using linear models. We show how linear models are represented using the image and basis file types. We illustrate the power of the representation by show-

---

*Present address: Center for Visual Science, 274 Meliora Hall, University of Rochester, Rochester, NY 14627.

†This statement is strictly true only when the image data are within the display gamut of both monitors. The problem of how to best render image data that lie outside of the gamut of a display device is a topic of active research. For two approaches, see Stone, Cowan, and Beatty[1] and Wandell and Brainard.[2]

ing how it facilitates estimation of surface reflectance functions from camera data.

The purpose of this article is to describe design principles for calibrated color image processing and is it not intended as a detailed description of our implementation of these principles. We describe our implementation, which we call the *Color Analysis Package* or *CAP*, in a separate technical report.[7] Both the technical report and the CAP software itself are available upon request.

Data Representation

General Considerations

In conventional color image processing, three numbers are used to represent color image data. These numbers are derived from two fundamentally different physical processes. When we display a color image, the numbers represent the intensities of the light emitted by three monitor phosphors. The phosphor intensities determine the spectral power distribution of the light emitted by the monitor. When we acquire a color image, the numbers represent the responses of the color camera sensors. The sensor responses are determined by the spectral power distribution of the light incident at the camera.

Common across the two physical processes is the presence of a light signal, which we call the *color signal*. When we display an image the color signal is the light emitted by the monitor. When we sense an image the color signal is the light incident at the camera. In both cases we represent the spectral power distribution of the color signal using the row vector $\mathbf{c}$, whose entries are the spectral power of the light at sample wavelengths.

Our image data structures are designed to make the relationship between the image data and the color signal explicit. Consider the linear equations that define the dependence between the color signal and image data in these two cases. In both cases, we represent the image data as a row vector, $\mathbf{d}$. When the image data represent monitor phosphor intensities, then $\mathbf{d}$ is three-dimensional and its entries are the red, green, and blue phosphor intensities. The relationship between the image data vector and the color signal vector is

$$\mathbf{c}_m = \mathbf{d}_m \mathbf{B}_m^T, \tag{1}$$

where we have used the subscript "m" to remind us that the equation refers to the light emitted by monitor phosphors. The three columns of the matrix $\mathbf{B}_m$ contain the spectral power distributions of the light emitted by the three monitor phosphors at peak output. The monitor color signal is always the weighted sum of these three spectral power distributions.[8,9]

Equation (1) separates the representation of the monitor image into two parts. The image data vector $\mathbf{d}_m$ describes the phosphor intensities at an image location. The matrix $\mathbf{B}_m$ describes the spectral properties of the monitor. When both $\mathbf{d}_m$ and $\mathbf{B}_m$ are included in the image representation, we can use Equation (1) to compute the vector $\mathbf{c}_m$ that describes the light that will be displayed to the observer.

When the image data represent color camera sensor responses, we can still represent the data using a three-dimensional row vector, $\mathbf{d}$. In this case the vector entries are the red, green, and blue camera sensor outputs. When $\mathbf{d}$ represents camera sensor responses, the relationship between the image data vector and the color signal vector becomes

$$\mathbf{d}_c = \mathbf{c}_c \mathbf{B}_c, \tag{2}$$

where we have used the subscript "c" to remind us that the equation refers to an acquired camera image. The image data vector $\mathbf{d}_c$ describes the camera responses at an image location. Like Equation (1), Equation (2) also separates the representation of the image into two parts. Here the three columns of the matrix $\mathbf{B}_c$ are the spectral responsivities of the three camera sensors. When both $\mathbf{d}_c$ and $\mathbf{B}_c$ are included in the image representation, we can use Equation (2) to compute the color signal vectors that are metameric to the incident color signal $\mathbf{c}_c$.

Implementation

Equations (1) and (2) show that for a complete specification of either a monitor or a camera image, we need to represent both a data vector $\mathbf{d}$ and a basis matrix $\mathbf{B}$. It is only the meaning of these entities that varies between the two types of image. Following the logic of Equations (1) and (2), we designed image processing software, which we call the Color Analysis Package or CAP, to represent images using two file types. One file, the *image file*, contains a data vector $\mathbf{d}$ for each of the spatial locations of the image. The second file, the *basis file*, contains the matrix $\mathbf{B}$.

The list of data vectors represented in the image file can be grouped to form a single matrix; each row of the matrix corresponds to the data vector from a single spatial location of the image. We call this data matrix $\mathbf{D}$. The row dimension of the matrix $\mathbf{D}$ is equal to the number of spatial locations in the image. The column dimension is equal to the column dimension of the data vector $\mathbf{d}$. For most monitor and sensor images the column dimension will be three. In this case the image file contains the data that are specified in most image processing systems.

Our representation is distinguished from conventional image representations by the inclusion of the spectral information recorded in the basis file. The basis file contains spectral information that permits us to relate the image data to the color signal. For example, in the case of a monitor image the basis file contains the spectral power distribution of the light emitted by the monitor phosphors at maximum intensity. In the case of a camera image the basis file contains the spectral responsivity of the camera sensors. The name of the basis file is stored at the beginning of the image file. Any program reading the image file, then, can also read the basis file and access the spectral information needed to perform calibrated color image processing.

Figure 1 shows an example of how image and basis files represent color data. Image files begin with a header. The header specifies the spatial properties of the image such as the number of spatial rows and columns. The header also specifies the column dimension of the data matrix, the name of the basis file, and other information. The data matrix **D** is stored following the header. Basis files also begin with a header. Because the matrix **B** always describes spectral properties, basis file headers contain spectral sampling information such as the first wavelength sample, the number of samples, and the sample spacing. The matrix **B** is stored following the header. We discuss our implementation in more detail elsewhere.[7]

It is worth noting that each of the standardized color representations can be divided into two classes that refer to monitor-like representations and sensor-like representations. The CIE spectral primary system is an example of a monitor-like representation. Equation (1) describes the CIE spectral primary representation when the entries of the image data vector are the CIE spectral tristimulus values, *RGB*, and the columns of **B** are spectral power distributions of the CIE spectral primaries. The CIE 1931 standard observer is an example of a sensor-like representation. Equation (2) describes the standard observer representation when the entries of the image data vector are the CIE tristimulus values, *XYZ*, and the columns of **B** are the color-matching functions $\bar{x}, \bar{y}, \bar{z}$ (see Wyszecki and Stiles, pp. 121–132[10]).

An Application

To illustrate how the CAP image representation facilitates calibrated color image processing, we will describe an example of how to render a camera image on a monitor. Suppose the camera image is stored in an image and basis file pair. Recall that in the image file we group the data vectors into a matrix **D**, where each row of the matrix is the vector describing the image data at a single location. Similarly, we group the color signals into a single matrix **C**. Each row of the matrix describes the color signal at a single spatial location. We can extend Equation (2), which describes the relationship between the incident color signal and the camera data at a single location, into a matrix equation that describes this relationship for all locations:

$$\mathbf{D}_c = \mathbf{C}_c \mathbf{B}_c \tag{3}$$

The image file contains the matrix $\mathbf{D}_c$ and the associated basis file contains the matrix $\mathbf{B}_c$. The subscript "*c*" reminds us that these are camera image data and that the color signal was incident at the camera.

Ideally, we would like to render the camera image data so that the color signal emitted by the monitor is equal to the color signal incident at the camera. Of course, monitors cannot emit arbitrary color signals. In rendering the image on a particular monitor, we are restricted to color signals of the form

$$\mathbf{C}_m = \mathbf{D}_m \mathbf{B}_m^T. \tag{4}$$

This equation generalizes Equation (1). The subscript "*m*" reminds us that these are monitor image data and that the color signal was emitted by the monitor.

Although it is not possible to reproduce the original color signal, we can create a monitor image which, when recorded by the original camera, will match the original sensor responses. This matching constraint is expressed by the matrix equation

$$\mathbf{D}_c = \mathbf{D}_m (\mathbf{B}_m^T \mathbf{B}_c). \tag{5}$$

We derive Equation (5) by noting that Equation (3) gives the camera sensor responses to any color signal, and in particular to the color signal $\mathbf{C}_m$ expressed in Equation (4). By substituting ($\mathbf{D}_m \mathbf{B}_m^T$) for $\mathbf{C}_c$ in Equation (3) and regrouping the matrix multiplications, we arrive at (5). Thus the matrix product on the right-hand side of Equation (5) calculates the camera sensor responses to the monitor image. When Equation (5) holds, these responses are identical to the original camera data. When the camera has three color

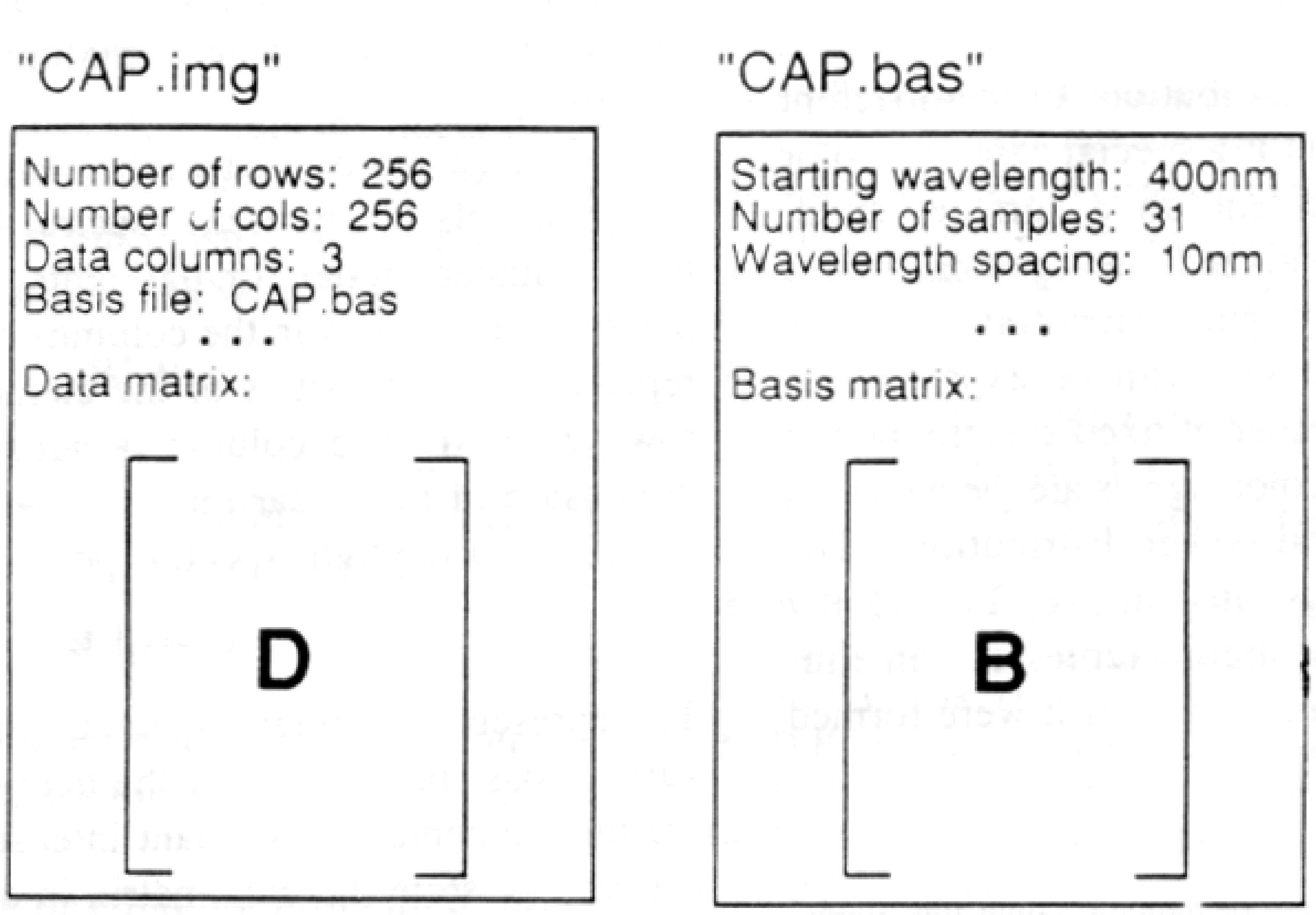

FIG. 1. In the CAP implementation we represent color image data using two file types. *Image files* contain the data matrix **D** for each of the spatial locations of the image. *Basis files* contain the matrix **B**.

sensors, and the monitor has three phosphors, the matrix $(\mathbf{B}_m^T\, \mathbf{B}_c)$ is a three by three matrix. If the matrix is nonsingular, we can invert it and solve Equation (5) for the monitor phosphor intensities

$$\mathbf{D}_m = \mathbf{D}_c\, (\mathbf{B}_m^T\, \mathbf{B}_c)^{-1}. \tag{6}$$

Equation (6) is a general procedure for rendering camera data.* The image and basis file representation makes the spectral properties of the camera and monitor explicit. An application program can use the representation and Equation (6) to render image data from any camera on any monitor. If the camera spectral responsivities are within a linear transformation of the human color-matching functions, the appearance of the displayed image is independent of the monitor's phosphor spectral power distributions.[4] If the camera sensor responsivities are not within a linear transformation of the color-matching functions, the appearance of the rendered image may not match the appearance of the original incident image. In this case we can use rendering procedures that take the human color-matching functions and restrictions on the color signal into account.[5]

Illuminant and Surface Representation

Linear Models

Calibrated color image processing may include explicit graphics models of the physical factors involved in image formation: color signals, illuminants, and surfaces. An example is rendering a digitally archived representation of a painting as it would appear under different illuminants. To include these physical factors in our image processing system, we must be able to represent and manipulate them. All of these factors are described by functions of wavelength. We have already developed one example of representing a functions of wavelength using CAP image and basis files. Our image and basis file representation of a monitor image represents the monitor color signal through Equations (1) and (4). Now we discuss the general representation of spectral functions.

Consider the spectral power distributions of the different phases of daylight. An illuminant is a special case of a color signal that happens to be incident on a surface. Judd, MacAdam, and Wyszecki[11] analyzed the spectral power distributions of a large set of daylights. They found that the daylight spectral power distributions can be expressed as the weighted sum of a small number of fixed spectral power distributions, just as the monitor color signals are the weighted sum of the fixed monitor spectral power distributions. The principle of their analysis is illustrated in Fig. 2.

The large circle drawn in the figure represents an illuminant. We can think of the illuminant as if it were formed by the mixture of a small number of *elementary illuminants*. When the illuminant is constrained in this way, we need not specify the full spectral power distribution of the illuminant in order to know its properties. If we know the spectral power distribution of the elementary illuminants, then from the intensity of each of the elementary illuminants we can compute the full illuminant spectral power distribution. When the measured illuminants can be modeled as the mixture of a small number of elementary illuminants, we say that the illuminants are described by a *linear model*. We call the number of elementary illuminants the *dimension* of the linear model.

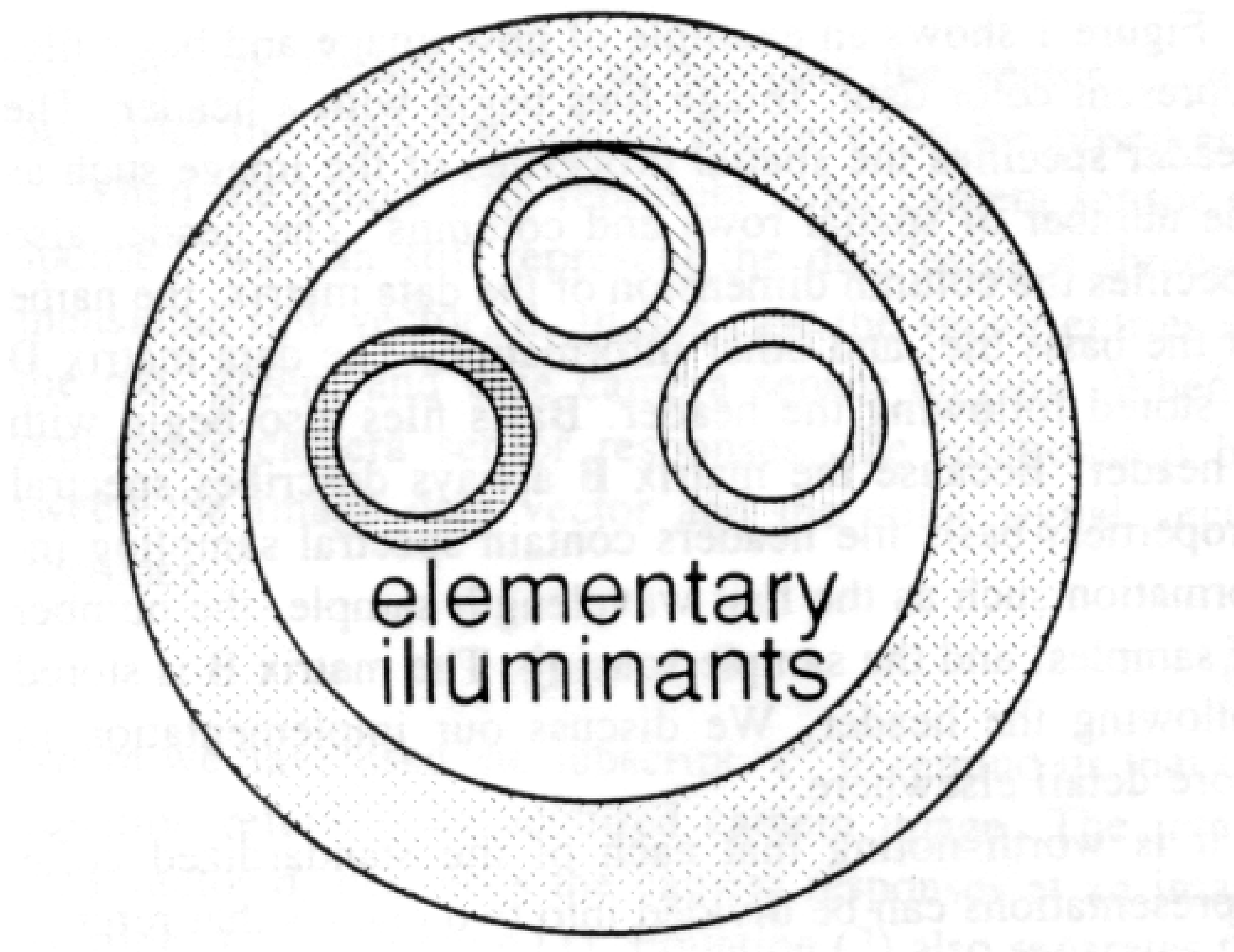

FIG. 2. Linear model for illuminants. The illuminant is formed by the mixture of three elementary illuminants. Each of these has a fixed relative spectral power distribution and varies only in its overall intensity.

In the case of the color signal from a monitor, the phosphor emissions play the role of the elementary illuminants. But in general the elementary illuminants are theoretical constructs and need not correspond to a physical process. They may have negative power values at some wavelengths. An elementary illuminant with negative power can never be seen directly, but only in a mixture with other elementary illuminants.

The image and basis file representation is designed to use linear models to represent spectral functions. Suppose we have a linear model for daylights. We represent the elementary illuminants in the columns of a basis matrix $\mathbf{B}$. We represent the intensities of the elementary illuminants in a row vector $\mathbf{d}$. The column dimension of $\mathbf{B}$ and $\mathbf{d}$ is the dimension of the linear model. If we let $\mathbf{e}$ be a row vector describing a daylight spectral power distribution, then

$$\mathbf{e} = \mathbf{d}\, \mathbf{B}^T. \tag{7}$$

To represent illuminant spectral power distributions that vary across space we form the data matrix $\mathbf{D}$, whose rows are the elementary illuminant intensities at each spatial location. We store the data matrix in an image file. We store the matrix $\mathbf{B}$ in a basis file.

The CIE has standardized a three-dimensional linear model for daylights (see Wyszecki and Stiles, p. 145[10]). In addi-

*Again, we do not address the problem of monitor gamut limitations. If some of the computed entries of $\mathbf{D}_m$ are negative or greater than the maximum displayable phosphor intensity, then the rendering procedure has generated an out-of-gamut data vector. See previous footnote.

tion, many investigators have studied the use of linear models to describe surface reflectance functions.[12–16] Their work indicates that a large number of natural surfaces may be accurately represented using linear models with between three and six dimensions. Of course, the image and basis file format allows surface reflectance functions to be represented using linear models.

An Application

We close by returning to the example of archiving a painting. We start with a camera sensor image of a painting illuminated by a known source. Our goal is to store the reflectance functions of the painting's surfaces. This would allow us to render our archived data as it would have looked under any illuminant. Suppose that the camera has three classes of sensors and that the painting's surface reflectance functions are well-described by a known three-dimensional linear model. Under these conditions, Buchsbaum[3] presents a method for recovering surface reflectance functions that is easily implemented as a CAP application program.

The spectral power distribution of the color signal incident at the camera is the wavelength-by-wavelength product of the illuminant spectral power distribution and the surface reflectance function. For uniformly illuminated surfaces, we can regard the illuminant as modifying the camera spectral responsivities (see Judd and Wyszecki, pp. 145–149[17]). Let $\mathbf{S}$ be a matrix whose rows are the painting's surface reflectance functions. Let $\mathbf{B}'_c$ be a matrix whose columns are the modified camera sensor responsivities. Each column of $\mathbf{B}'_c$ is obtained from the true camera sensor responsivity through wavelength-by-wavelength multiplication with the illuminant. The relation between the surface reflectance functions and the camera sensor data matrix is given by

$$\mathbf{D}_c = \mathbf{S}\,\mathbf{B}'_c. \tag{8}$$

We express the surface reflectance functions with respect to the known linear model:

$$\mathbf{S} = \mathbf{D}_s\,\mathbf{B}_s^T. \tag{9}$$

The three columns of the matrix $\mathbf{B}_S$ contain the linear model for the painting's surface reflectance functions and the rows of the matrix $\mathbf{D}_s$ contain the unknown model weights for each surface. Combining Equations (8) and (9) we find

$$\mathbf{D}_c = \mathbf{D}_s\,(\mathbf{B}_s^T\,\mathbf{B}'_c). \tag{10}$$

If the three by three matrix $(\mathbf{B}_s^T\ \mathbf{B}'_c)$ is nonsingular, then we can invert it and estimate the model weights from the camera sensor responses

$$\mathbf{D}_s = \mathbf{D}_c\,(\mathbf{B}_s^T\,\mathbf{B}'_c)^{-1}. \tag{11}$$

The model weights together with the linear model determine the painting's surface reflectance functions through Equation (9).

Equation (11) provides a procedure for estimating surface reflectance functions. Because the image and basis file representation makes the spectral properties of the illuminant, surfaces, and camera explicit, an application program can take advantage of this procedure. If data from a camera with more than three classes of sensors can be obtained, the same general procedure can be used to obtain more precise estimates.

Summary and Discussion

We have described the design philosophy and image representation of a calibrated color image processing system. The design emphasizes the representation of spectral functions necessary for device-independent display and capture of color data. We have implemented these principles in a software package that we have used in our laboratory for several years. We have found that representing the physical factors that underly color imaging and representing these factors with respect to linear models are both useful principles for designing color image processing software. These basic design principles can be incorporated into a new generation of image processing software that integrates both spatial and spectral image processing.

Acknowledgement

This work was supported by Grant No. 2 RO1 EY03164 from the National Eye Institute, and Grant NCC 2-307 from the NASA-Ames Research Center. Some of the material in this article is based upon work supported under a National Science Foundation Graduate Fellowship. The opinions, findings, and conclusions of this paper do not necessarily reflect the views of the National Science Foundation. The authors thank C. McConnell for help with design and programming.

1. M. Stone, W.B. Cowan, and J.C. Beatty, "Gamut mapping and the printing of digital color images," *ACM Trans. Graphics*, **7**(4), 249–292, (1988).
2. B.A. Wandell and D.H. Brainard, "Towards cross-media color reproduction," *Proceedings OSA Applied Vision Topical Meeting*, San Francisco, CA, July 12–14, 1989.
3. G. Buchsbaum, "A spatial processor model for object color perception," *J. Franklin Institute*, **310**(1), 1–26, (1980).
4. B.K.P. Horn, "Exact reproduction of colored images," *Computer Vision, Graphics and Image Processing*, **26**, 135–167 (1984).
5. B.A. Wandell, "Color rendering of camera data," *Color Res. Appl., Supplement*, **11**, S30–S33, (1986).
6. B.A. Wandell, "The synthesis and analysis of color images," *IEEE Trans. PAMI*, **PAMI-9**, 2–13, (1987).
7. D.H. Brainard and B.A. Wandell, "The color analysis package," *Stanford Applied Psychology Laboratory Technical Report*, no. 89–3, 1989.
8. W.B. Cowan and N. Rowell, "On the gun independence and phosphor constancy of colour video monitors," *Color Res. Appl., Supplement*, **11**, S33–S38, (1986).
9. D.H. Brainard, "Calibration of a computer controlled color monitor," *Color Res. Appl.*, **14**, 23–34, (1989).
10. G. Wyszecki and W.S. Stiles, *Color Science*, John Wiley and Sons, New York, 1982.
11. D.B. Judd, D.L. MacAdam, and G.W. Wyszecki, "Spectral distribution of typical daylight as a function of correlated color temperature," *J. Opt. Soc. Am.*, **54**, 1031, (1964).

12. J. Cohen, "Dependency of the spectral reflectance curves of the Munsell color chips," *Psychon. Sci.* **1**, 369–370, (1964).
13. L.T. Maloney, "Evaluation of linear models of surface spectral reflectance with small numbers of parameters," *J. Opt. Soc. Am. A*, **3**(10), 1673–1683, (1986).
14. J.P.S. Parkkinen, J. Hallikainen, and T. Jaaskelainen, "Characteristic spectra of Munsell colors," *J. Opt. Soc. Am.*, **6**(2), 318–322, (1989).
15. E.L. Krinov, "Surface reflectance properties of natural formations," *National Research Council of Canada: Technical Translation*, **TT-439**, (1947).
16. T. Jaaskelainen, J. Parkkinen, and S. Toyooka, "A vector-subspace model for color representation," *J. Opt. Soc. Am. A*, (*in press*, 1989).
17. D. Judd and G. Wyszecki, in *Color in Business, Science, and Industry*, John Wiley & Sons, New York, 1975.

Received August 30, 1989; accepted May 1, 1990