PROGRESSIVE TRANSMISSION OF PSEUDO-COLOR IMAGES

by

Andrew C. Hadenfeldt

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Electrical Engineering

Under the Supervision of Professor Khalid Sayood

Lincoln, Nebraska

December, 1991

PROGRESSIVE TRANSMISSION OF PSEUDO-COLOR IMAGES

Andrew C. Hadenfeldt, M.S.

University of Nebraska, 1991

Adviser: Khalid Sayood

The transmission of digital images can require considerable channel bandwidth. The cost of obtaining such a channel can be prohibitive, or the channel might simply not be available. In this case, progressive transmission of the image data can be useful. Progressive transmission presents the user with a coarse initial image approximation, and then proceeds to refine it. In this way, the user tends to receive information about the content of the image sooner than if a sequential transmission method was used. The result is that the user can determine the image content more quickly, allowing early termination of transmission if the image is to be rejected. Progressive transmission finds application in image database browsing, teleconferencing, medical and other applications.

In this thesis, a progressive transmission scheme is developed for use with a particular type of image data, the pseudo-color or color-mapped image. Such images consist of a table of colors called a colormap, plus a two-dimensional array of index values which indicate which colormap entry is to be used to display a given pixel. This type of image presents some unique problems for a progressive transmission coder, and techniques for overcoming these problems are developed. A computer simulation of the color-mapped progressive transmission scheme is developed to evaluate its performance. Results of simulation using several test images are presented.

Progressive Transmission of Pseudo-Color Images

Table of Contents

ACKNOWLEDGEMENTS i

LIST OF FIGURES ii

LIST OF TABLES iv

LIST OF ABBREVIATIONS v

CHAPTER 1. INTRODUCTION 1

CHAPTER 2. PROGRESSIVE IMAGE TRANSMISSION 4

2.1 Introduction to Progressive Transmission 4

2.2 Types of Progressive Transmission Schemes 8

2.3 Implementing Progressive Transmission 10

2.4 Examples of Progressive Transmission from the Literature 10
2.4.1 Transform-Based PIT Schemes
2.4.2 Non-Transform PIT Schemes

CHAPTER 3. COLOR AND DIGITAL IMAGES 17

3.1 Basic Color Spaces 18
3.1.1 CIE RGB Space
3.1.2 CIE XYZ Space
3.1.3 NTSC RGB Space
3.1.4 NTSC YIQ Space

3.2 Chromacticity 22

3.3 Perceptual Color Spaces 22
3.3.1 CIE Uniform and Modified Uniform Chromaticity Scales
3.3.2 CIE L*a*b* Perceptual Color Space
3.3.3 CIE L*u*v* Perceptual Color Space
3.3.4 Other Perceptual Color Systems

3.4 Digital Image Display Hardware 30
3.4.1 Achromatic Image Displays
3.4.2 Full-Color Image Displays
3.4.3 Pseudo-Color (Color-Mapped) Image Displays

CHAPTER 4. COLOR MAPS FOR IMAGE DISPLAY 34

4.1 Selecting Colormap Entries 34
4.1.1 Popularity Algorithm
4.1.2 Median-Cut Algorithm
4.1.3 Variance-Based Color Quantization
4.1.4 Iterative Clustering Algorithms

Table of Contents (continued)

4.2 Converting the Image to Use the Colormap 39

4.3 Source Images for this Work 40

4.4 Properties of Color-Mapped Images 40

4.5 *Colormap Sorting* 43
4.5.1 Greedy Sorting Algorithm
4.5.2 Sorting Using Simulated Annealing

CHAPTER 5. A PT SCHEME FOR COLOR-MAPPED IMAGES 55

5.1 A Basic CMPT Framework 55

5.2 Objective Evaluation Criteria 56

5.3 Completing the CMPT System 57
5.3.1 Choosing Representative Values
5.3.2 Transmitting the Pyramid Levels
5.3.3 Encoding the Pyramid Nodes
5.3.4 Encoding the Node Selection Bits

5.4 Simulation Results 68
5.4.1 Results Using an RGB-Based Information Measure
5.4.2 Results Using an L*u*v*-Based Information Measure
5.4.3 *Comparison of Lossless Transmission Rates*

5.5 Possible Modifications and Improvements 74

CHAPTER 6. SUMMARY AND CONCLUSIONS 81

REFERENCES 83

Acknowledgements

I am grateful to my committee members, Dr. A. John Boye and Dr. Robert Maher, for their participation and helpful suggestions. Thanks also goes to Dr. Donald Rundquist of CALMIT for supplying the geographical test images in this work. Appreciation is also given to Marjorie Bisbee of the Engineering Research Unit, for her help with printing the color figures. Special thanks goes to my adviser, Dr. Khalid Sayood, for his guidance and friendship. Finally, I thank my parents for providing me with the opportunity to better myself through education.

This work was supported by the NASA Goddard Space Flight Center under grant NAG 5-916

List of Figures

Figure		Page
2.1	Sequential Scan Image Transmission	6
2.2	Progressive Image Transmission	7
2.3	Progressive Pass vs. Rate	8
2.4	Pyramid Data Structure	9
3.1	1931 CIE RGB Color-Matching Functions	19
3.2	1931 CIE XYZ Color-Matching Functions	20
3.3	CIE Chromaticity Diagram	23
3.4	Weber's Law of Contrast Dependency	24
3.5	Just-Noticeable Color Differences	26
3.6	A Model For Color Vision	28
3.7	Achromatic Frame Buffer	30
3.8	Full-Color Frame Buffer	31
3.9	Color-Mapped Frame Buffer	33
4.1	Hat, Full-Color Original	50
4.2	Hat, 8-bit Color Map	50
4.3	Park, Full-Color Original	51
4.4	Park, 8-bit Color Map	51
4.5	Omaha, Full-Color Original	52
4.6	Omaha, 8-bit Color Map	52
4.7	Lincoln, Full-Color Original	53
4.8	Lincoln, 8-bit Color Map	53

List of Figures (continued)

4.9	Original and Sorted Color Maps, Park Image	54
4.10	Park, Original and Sorted Color Map, 5 bits/pixel	54
5.1	Block of Pyramid Nodes (2×2)	64
5.2	CMPT Hat Pass 0, 0.16 bpp	78
5.3	CMPT Hat Pass 1, 0.40 bpp	78
5.4	CMPT Hat Pass 2, 1.06 bpp	79
5.5	CMPT Hat Pass 3, 2.21 bpp	79
5.6	CMPT Hat Pass 4, 3.42 bpp	80
5.7	CMPT Hat Pass 5, 4.54 bpp	80

List of Tables

Table		Page
4.1	Entropies of the Source Images	42
4.2	Resultant Images With Circularly Sorted Colormaps	48
4.3	Resultant Images With Linearly Sorted Colormaps	48
5.1	CMPT Using An RGB-Based Information Measure	70
5.2	CMPT Using An L*u*v*-Based Information Measure	72
5.3	Lossless Coding Comparison	73

List of Abbreviations

CIE	*Commission Internationale de l'Eclairage* (International Committee on Illumination)
CMPT	Color-mapped progressive transmission
DCT	Discrete cosine transform
FOH	First-order hold
HVS	Human visual system
L*a*b*	A CIE perceptual color space
L*u*v*	A CIE perceptual color space
MSE	Mean-squared error
NSB	Node-selection bit
NTSC	National Television Standards Committee
PT/PIT	Progressive (image) transmission
RGB	Red-green-blue (a color space)
SPOT	*Le Systemie Pour l'Observation de la Terre*, a French earth-observing satellite
TMS	Thematic Mapping System, an earth-observing satellite sensor
UCS	CIE Uniform Chromaticity Scale
VQ	Vector Quantizer
YIQ	Luminance-chrominance (NTSC transmission color space)
ZOH	Zero-order hold

1. Introduction

Digital images are becoming increasingly more important for modern communications. Presenting data in a visual form takes advantage of the highest bandwidth, most complex sensor available to the user, the human eye. Images are popular for use in scientific applications, where they provide a means for the researcher to visualize large volumes of data quickly and easily. Images have become indispensable for communications in the business world, where graphical presentations and fax machines are commonplace. With such innovations as local-area networks (LANs), the proposed Integrated Services Digital Network (ISDN), and the introduction of high-definition television (HDTV), images will play an even greater role in our work and recreational lives.

One particularly interesting area in which images will be incorporated is database applications. Today large image databases already exist, containing several terabytes (10^{12} bytes) of data gathered from the earth and other parts of the solar system. The need for efficient methods of storage is obvious. More importantly, techniques must be devised to allow users to access this data. One can easily conceive an image database browsing application. A remote user isolates a group of images using conventional query methods, and subsequently wishes to visually search through the group for images of particular interest. For example, if a user is interested in a particular geographical feature, he or she desires a view of the feature that is not obscured by clouds. In this single-host, many-user scenario, the computing power and communication link available to the users may be limited. Since image transmission can require a considerable amount of channel bandwidth, a user browsing through only a few images may spend hours waiting for the data to be transferred to a local display. It is this type of situation for which

progressive image transmission is intended. Progressive transmission presents the user with a coarse reproduction of the original image, and then proceeds to refine it. The initial approximation might be available to the user in as little as one second. During refinement, the user can view the approximations. Getting the information in front of the user quickly allows faster recognition of the image content. Thus, the user can determine whether to accept or reject the image sooner, and can avoid wasting channel resources on images which are of no use to the user. Progressive transmission also finds uses in teleconferencing, medical and military applications.

Many forms of image data exist. For some applications an achromatic ("black-and-white") image is sufficient. Other applications, however, may benefit from or require the use of color image data to be effective. In medical applications, color is often used to indicate areas of interest in X-ray or MRI images. Multispectral satellite data is often presented in a color format, by assigning a red, green, or blue wavelength to a channel. A large amount of data may be associated with this type of full-color image, often several megabytes. Displays which can accurately display these images are also expensive. Instead, modern personal computers and workstations often use a pseudo-color, or color-mapped, display system. In this type of display, the values stored in the display memory are use as indices into a color table (called the colormap) to determine the color of the pixel on the CRT. To allow the use of this display equipment, full-color image data is often reduced to a colormap plus a two-dimensional color index array. This is the form of the image data which was used for this thesis.

The remainder of this thesis describes the development of a progressive transmission scheme for color-mapped images. In Chapter 2, general issues concerning progressive transmission are presented, along with several schemes from the literature. Chapters 3 and 4 provide background material on how humans perceive color and how color-mapped images might

be created from their full-color representations. Since this image data will be encoded by a progressive transmission scheme, some analysis of the properties of color-mapped image data is also presented. In particular, color-mapped images present some unique coding problems, and methods of overcoming these difficulties are discussed. Chapter 5 describes the completed system for color-mapped progressive transmission (CMPT). Results of computer simulations of the CMPT coder are presented. Finally, Chapter 6 presents a summary of the results of this thesis.

2. Progressive Image Transmission

In this chapter, the technique of progressive image transmission (PIT) is introduced. Motivation for using this technique is also discussed. Different methods of progressive transmission are found in the literature, and several examples of these techniques are presented. Many of the concepts in this chapter will be used in Chapter 5 to construct the progressive transmission coder developed for this thesis.

2.1 Introduction to Progressive Transmission

As an example, consider a user of a remote image database where access is obtained via a modem. The user may use standard database query techniques to isolate images of interest. At that point, the user may wish to view the images to see if they satisfy his or her needs. The user requests that a copy of the images be transmitted to a local display.

Many techniques for transmitting single-frame digital images work in the following manner. First, the image frame is encoded (compressed) using some algorithm. Next, the compressed data is transmitted over the channel. Finally, the receiver reconstructs the entire image from the channel data, possibly with some distortion. This technique is sufficient for many applications. However, if the channel bandwidth is low, transmitting even the encoded (compressed) data may require a considerable amount of time. During this time, the user at the receiver may need to wait for all of the encoded data to be received before the image can be decoded and used.

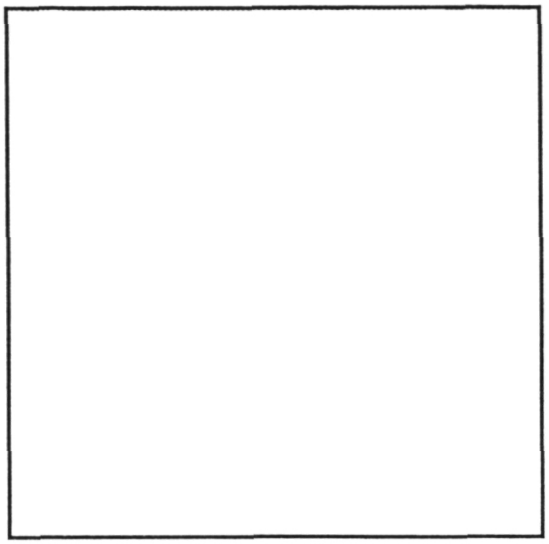
A somewhat better situation is illustrated in Figure 2.1. In this scheme, the transmission technique is to send the image pixels row by row, from top to bottom. This allows the image to be displayed during transmission. However, a typical user will not be able to recognize the

received image until one-third to one-half of the image has been received. For example, if the image in the figure is 512×512 pixels, 8 bits/pixel, and is transmitted using a 9600 bits/sec line, a total of 3.64 min. will be required to transmit the entire image. The user will need to wait 1.2-1.8 min. before the image is usable. The user's attention becomes focused on the scanning process, and he or she quickly becomes fatigued.

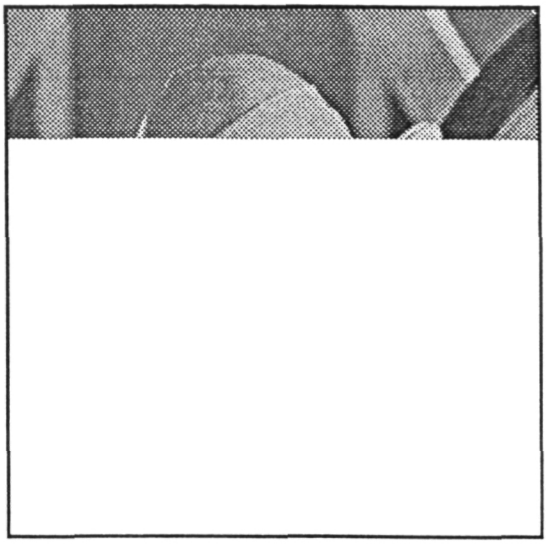
In such a case, progressive image transmission (PIT) is useful. Progressive transmission uses a multiple-pass transmission method to send the image. An example of this technique is shown in Figure 2.2. The amount of time required to transmit each image is shown, based on the conditions of the previous example. A PIT system first presents the user with a coarse approximation of the image, then proceeds to refine it. The user is able to view the received image while transmission is in progress. The result is that recognition of image content will often occur more quickly [2][3]. Progressive transmission finds use in remote image database applications, photojournalism, teleconferencing, low-bandwidth military applications, and medical applications.

Figure 2.3 illustrates some properties of a typical progressive transmission scheme for an 8-bit image. The graph shows the average amount of data required in bits/pixel to obtain each progressive image approximation. The amount of data required for early passes is very low, allowing the user to quickly receive a representation of the image. If an interactive user decides to terminate transmission early, a large amount of data need not be sent. This property of a PIT system is known as *effective compression.*

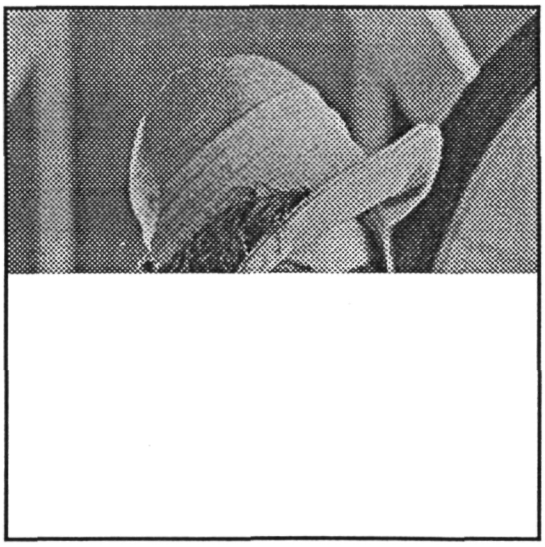
Figure 2.3 also illustrates a potential disadvantage of progressive transmission. If the transmission process is allowed to complete, the total amount of data needed to transmit the image is now greater than what would be required to transmit the original image without PIT. The requirement that the coder be able to produce several image approximations during transmission



1 second



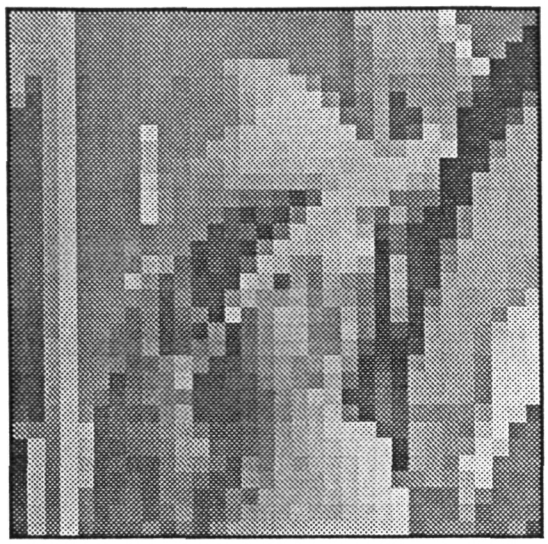
55 seconds



1.8 minutes

3.6 minutes (complete)

Figure 2.1: Scanline Sequential Image Transmission



1 second

55 seconds

1.8 minutes

3.6 minutes (complete)

Figure 2.2: Progressive Image Transmission

often results in decreased compression performance. Fortunately, this problem can be overcome to some extent, as the examples in Section 2.4 show.

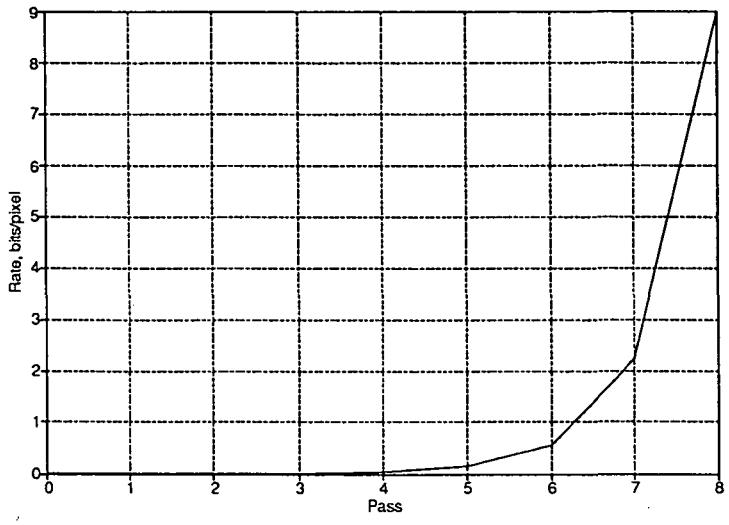


Figure 2.3: Progressive Pass vs. Rate

2.2 Types of Progressive Transmission Schemes

Progressive transmission schemes may be divided into two categories: those that use transform coding [1], and those that do not. Transform-based coders are usually lossy coders, i.e., the original image is not reproduced without distortion at the receiver. Other types of coders are usually lossless. However, it is possible to make a lossless coder into a lossy one by early termination of the progressive transmission algorithm, and it is also possible to make a lossy coder into a lossless one by adding some type of lossless residual error coder as a final step.

For the schemes that use some form of transform coding, the discrete cosine transform (DCT) [4] is most popular. The DCT does a good job of compacting energy into only a few components; thus, a good representation of an image may be obtained by retaining only a few coefficients of the transform. Also, a progressive transmission scheme is easy to implement by transmitting only a few of the DCT coefficients at a time. In addition, fast algorithms for

computing the DCT exist, and hardware processors which perform this operation are available on a single chip.

For schemes that do not use some sort of transform coding, there are many variations. Nearly the only thing they have in common is the ability to generate coded data in progressive stages which can be used to reconstruct the source image (possibly with some distortion, which hopefully is not too apparent). One popular technique is to use some type of "pyramid" structure to represent the source, an example of which is the quadtree structure. Figure 2.4 illustrates this structure.

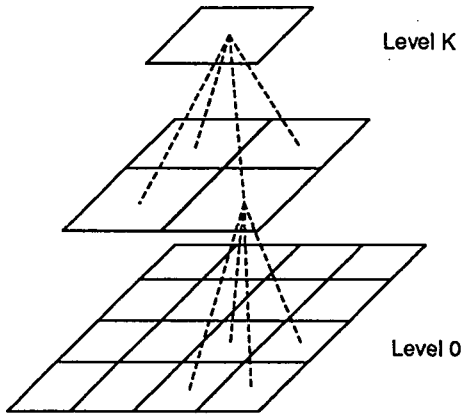


Figure 2.4: Pyramid Data Structure

In a pyramid coding scheme, successively lower-resolution versions of the source image are created according to some algorithm. In the figure, level 0 contains the original image. Each level contains one-fourth the amount of data as the previous level (the dimensions of the previous level are halved). The top level of the pyramid is typically a single value which represents the entire image in some way. Progressive transmission is achieved by transmitting the levels of the pyramid. Since most of the non-transform schemes are lossless coders, these coders typically require higher channel data rates than the lossy (e.g., transform) coders.

Some of the schemes attempt to improve the perceptual quality of early passes by

compensation based on a model of the human visual system (HVS). The compensation is typically aimed at reducing the bit rate while introducing distortion in ways which the eye cannot detect easily.

2.3 Implementing Progressive Transmission

As a general rule, transform-based schemes tend to be more complex from an implementation standpoint. These schemes tend to require more storage space for both encoder and decoder, and usually involve many floating-point calculations. Non-transform schemes may also require a significant amount of floating-point computation, but often involve only a simple difference or averaging operation. A scheme which uses some type of pyramid structure may also require a significant amount of memory, since the pyramid structure typically must be completely constructed at the transmitter before early passes can be sent to the receiver. (Note that this is not always true, since the progressive transmission scheme developed in this thesis uses a pyramid structure which requires little additional memory.) The availability of digital signal processing (DSP) chips and cheap memory devices may lessen the significance of these problems.

Since many of the schemes divide the source image into blocks (e.g, 8×8 or 16×16), some form of parallel processing should be possible. It is also possible that many schemes could be reworked slightly to a more parallel form.

2.4 Examples of Progressive Transmission from the Literature

Early work for progressive transmission appeared around 1979 [2], as a result of research into the use of hierarchial data structures in image processing applications. In this section, several methods of achieving PIT found in the literature are presented. Surveys of progressive transmission techniques are also provided by Tzou [5], and Wang and Goldberg [6].

2.4.1 Transform-Based PIT Schemes

Ngan [7] discusses five methods of transmitting the coefficients of an 8×8 DCT block

from the perspective of achieving the best perceptual quality with the fewest number of transmitted coefficients. The scheme found to work best uses a distortion measure to determine the optimal transmission sequence for the AC coefficients. Coefficients are ranked according to the mean-squared error of the received image when only that coefficient is transmitted together with the DC coefficient. While this method performs better than the others, it requires additional system overhead which the remaining four methods do not. Of these, a "zig-zag" transmission sequence performs best. This sequence sends the low spatial frequency DCT coefficients first, and proceeds by sending higher and higher frequency coefficients. The described research is not considered to be a complete progressive transmission scheme, since implementation issues such as coefficient quantization and bit allocation strategies are not discussed. However, the zig-zag transmission sequence is used as a part of some of the other complete schemes below.

Dubois and Moncet [8] code NTSC color images using a 16×16 DCT in several forms. One scheme, which is designed for progressive transmission, sorts the DCT coefficients into groups according to their properties and energies. These groups are then uniformly quantized and progressively transmitted using a Huffman code specifically designed for each group. Also included in the Huffman codebook are symbols representing run-lengths of zero coefficients, since there are usually many small amplitude coefficients in a typical block. In this scheme, the quantizer step size is used to control the coder data rate (at the expense of increased coefficient errors). A variation of the scheme sends each coefficient group using the same Huffman code. Both schemes give approximately the same performance, although the latter is less complex. In a third variation of the scheme, a lower bit rate is achieved by taking advantage of the properties of the human visual system. In this case, high-frequency transform coefficients are quantized with a larger quantizer step size, on the assumption that at least some of this increased high frequency noise (error) would be filtered out by the eye. For this last scheme, a reduction in coding rate of

24-33 percent over the first two schemes is reported, without a significant increase in visible distortion. The system is of medium complexity due to the cosine transform, but otherwise should not be difficult to implement. The amount of overhead information needed for the system is small. It should also be possible to improve the performance of this system by more carefully encoding the DCT coefficients (e.g., nonuniform quantization, simple vector quantization) without greatly increasing this complexity.

Chen and Pratt [9] use the DCT on 16×16 pixel blocks, for monochrome and NTSC color (YIQ format) images. DCT coefficients (scanned in the zig-zag manner described above) whose magnitudes exceed a given threshold are transmitted; all others are set to zero. To transmit a coefficient, the difference between the coefficient and the threshold is quantized and transmitted to the receiver using a Huffman code. A data buffer is used to provide a fixed data rate to the channel, and is constantly monitored to prevent overflow. If the buffer becomes too full, the number of bits used to quantize the DCT coefficients is reduced. Since many coefficients of each DCT block are zero, a second Huffman code is employed to perform a run-length coding for strings of zero coefficients. For color images, DCT coefficients for the Y, I, and Q components are transmitted in a similar manner. A hardware implementation of this system exists which can transmit NTSC video at 1.5 Mbits/sec. Although this system was not originally designed to operate in a progressive manner, it can be (and has been) modified to perform progressive transmission at low data rates.

Chitprasert and Rao [10] use an 8×8 DCT system based on that of Chen and Smith [11], with some modifications. The DCT coefficients are classified by their energies, and weighted by a model of the human visual system (HVS), to determine order of transmission to the receiver. Their claim is that the HVS weighting improves the perceptual quality of early coding stages, although an objective measure (SNR) indicates a slight decrease in image quality. The result is

a "classification map" and a "transmission map" which must be sent as overhead to the receiver for each frame. Quantization of the AC DCT coefficients is performed using a normalized, nonuniform Laplacian quantizer. Therefore, standard deviation matrices must also be transmitted as additional overhead to the receiver. Total overhead is reported to be approximately 0.07 bits/pixel, with "good" quality output achieved at a total rate of 0.3-0.5 bits/pixel. Complexity of this system is somewhat high due to the DCT and classification schemes.

2.4.2 Non-Transform PIT Schemes

Frank, Daniels and Unangst [12] develop a lossless "growth-geometry" scheme, which initially is used only for bilevel images. The scheme uses "seed" pixels, and "grows" portions of the images around them according to a selected rule. The growth rule is chosen from a predetermined set of rules. Encoding is essentially a pattern recognition task, to locate appropriate seed pixels and select a growth type. A small hardware implementation of the scheme exists for bilevel images. Suggestions are offered for extending the scheme to multilevel images (e.g., encoding each bit plane as a single bilevel image), but the potential for good multilevel image compression and a low complexity implementation seems small, especially for natural scenes.

Knowlton [3] divides the source image using a binary tree structure. A large rectangular block approximation of the image is progressively halved and refined, and eventually reproduces the original image losslessly. Operation of the system is independent of the characteristics of the source image, and the computational requirement is linear in the number of source image pixels. This should simplify a hardware implementation. The compression rate for this scheme is dependent upon the number of possible gray levels in the source image, 16 for the described research. The scheme has also been applied to bilevel images with good results. One drawback of the system is that early coding passes are more coarse than other schemes, requiring more passes to obtain a recognizable scene. In addition, good compression may be difficult for images

with a much larger number of gray levels, say 256 instead of 16, as the amount of encoder data would approximately double in this case. A modification of this scheme by Hill, et al. [13], has been used to implement an interactive image query system.

Sloan and Tanimoto [2] use a pyramid structure to losslessly encode the source image. With a pyramid scheme, progressive transmission is achieved by transmitting the levels of the pyramid. The lowest level of the pyramid is the N×N source image. The next pyramid level has dimension N/2×N/2, etc., and the top pyramid level is 1×1. Various methods for building and transmitting the pyramid are discussed. In one method, each successive pyramid level is formed by simply using the upper left-hand pixel of a 2×2 block from the previous level. This eliminates the need to transmit this one pixel again, and requires no computation. Then, only three of the four pixels need to be transmitted in the next pass. In another, the sum of the four pixels is transmitted instead. This also requires only three additional pixels during each successive pass, and little computation. The resultant image is lossless in both cases. A drawback of the pyramid approach is that the entire pyramid must be built before any data can be transmitted to the receiver, resulting in a transmission delay. However, the computational load for the scheme is low, and should minimize this delay. Compression does not appear to be as good as some of the other schemes here, but there are several areas of possible improvement. For example, a Huffman coder might be useful to encode the pixels and reduce the data rate.

Adelson and Burt [14] use a "Laplacian Pyramid" structure for representing the source image. First, a "Gaussian Pyramid" is formed, using a 5×5 convolutional operator with a Gaussian-like shape. The operator is applied recursively to form low-pass filtered reduced resolution pyramid levels. Each level is approximately 1/4 the resolution of the previous level. The Laplacian pyramid is formed by subtracting adjacent levels of the Gaussian pyramid. The Laplacian pyramid is then quantized to reduce its entropy and transmitted by levels to the receiver.

At the receiver, the Gaussian pyramid is reconstructed from the Laplacian pyramid. The receiver uses the same 5×5 convolutional operator to interpolate the received Gaussian pyramid levels to full size for the viewer. With this scheme good quality results are reported at rates of 0.7-1.6 bits/pixel.

Hoffman and Troxel [15] also use a pyramid-like, iterative technique to transmit an image. The various approximations are formed by repeatedly low-pass filtering and subsampling the image data. These approximations are then transmitted using a two-dimensional predictive coding technique. An interesting feature of this technique is that the displayed size of the received images is not fixed. The authors estimate an "optimal viewing size" for the received image, based on a model of the human visual system.

Dreizen [16] develops a lossless progressive transmission scheme using a pyramid structure. In this case, the pyramid structure is used along with an information measure to identify which portions of the image to develop first. A differential predictive coder, followed by a Huffman coder, is used to update the image at the receiver. Compression for the scheme ranges from 13-37 percent for the 128×128 8-bit images used. Complexity of the scheme is low. An added advantage of this scheme is that although a pyramid structure is used, the scheme does not require a large amount of memory to store it. Many of the features of this scheme were useful in designing the progressive transmission coder developed in this thesis.

Wang and Goldberg [17] also use pyramid structures to represent the source image. Two pyramids are created: a mean pyramid (consisting 2×2 pixel averages) and a difference (error) pyramid. Most of the coding effort goes toward coding the difference pyramid, which is used to refine the output at the receiver at each stage of transmission. Coding for the difference pyramid uses an LBG vector quantizer [18] designed exclusively for each portion of the difference pyramid. Quantizer errors are retained by the system and recoded on later passes. As a final

pass, an entropy code is used to transmit the residual errors, yielding a lossless image at the receiver. This approach yields a low data rate, but is very complex and computationally intensive. In this form, a hardware implementation of the scheme would be difficult. A possible modification to the scheme to reduce complexity might be to use a lattice VQ instead.

Wang and Goldberg [6] also offer a comparison of seven progressive transmission schemes which use various pyramid data structures. Among these are the mean and difference pyramids used in their previous work, and the Gaussian-Laplacian structure developed by Adelson and Burt. The schemes are compared using several criteria. Their choice for best performance is the reduced-difference (RD) pyramid [19], developed by the authors. The RD pyramid is formed from the mean pyramid, by forming differences between nodes on the same pyramid level. Only enough data to reconstruct the original difference pyramid need be retained. At the receiver, a Gaussian interpolation function is used to improve the quality of the received images, especially for early transmission passes. Their results indicate acceptable results are obtained at rates of 0.35-1.3 bits/pixel.

This chapter presents background information which is used to devise the progressive transmission scheme presented in Chapter 5. Before designing any coding scheme, it is useful to possess an understanding of its input data, so that its structure may be exploited. Chapters 3 and 4 provide this background for the color images used in this work.

3. Color and Digital Images

Color is defined as "that aspect of visual perception by which an observer may distinguish differences between two structure-free fields of view of the same size and shape, such as may be caused by differences in the spectral composition of the radiant energy concerned in the observation" [20]. Attempts to describe and quantify color have a long history, dating back several hundred years. In 1929, Munsell [21] introduced a color system which was used for many applications, consisting simply of a catalog of standard color "patches" which could be used to match colors. More precise measures have evolved since then, though it is interesting to note that perhaps the most significant theory concerning color perception was proposed over one-hundred years before Munsell's time. In 1802, Thomas Young proposed a tristimulus theory of color [22], stating that any color can be matched (perceptually) by an additive mixture of three colored lights, called *color primaries.* As we will see, these color primaries are not even required to be physically realizable sources. They must, however, be "linearly independent," i.e., it should not be possible to represent one of the primaries with a linear combination of the remaining two. Young's theory is supported by Maxwell and others who have determined that there are three different types of cones in the human retina, each receptive to a different band in the visible spectrum. An important consequence of this theory is that it is not necessary to exactly match the spectral distribution of a light source in order to create the same perceived color. The science of color and its measurement is known as *colorimetry*, and the following is a review of some of its important results. It will also be useful to take a brief look at how modern image display hardware presents image data to the user.

3.1 Basic Color Spaces

Given that three independent primaries are required to describe an arbitrary color, we must next choose such a set. A given set of primaries defines a basis set for a three-dimensional *color space*. The coordinates of a given color are known as the *tristimulus values* for that color. Several color spaces have come into common use, each having desirable characteristics. Some of these are described below.

3.1.1 CIE RGB Space

An attempt to standardize the choice of color primaries was made in 1931 by the *Commission Internationale de l'Eclairage* (CIE), the International Committee on Illumination. The three primaries chosen were monochromatic sources with wavelengths of 700.0, 546.1, and 435.8nm. These primaries fall in the red, green, and blue portions of the visible spectrum, and so is known as the CIE RGB color space. Along with the primaries, the CIE specifies a set of *color-matching functions* describing the relative contributions of the primaries required to match a (visible) monochromatic light of a specified wavelength, obtained from experimental observation. That is, the primaries R, G, and B may be combined to produce a color C using weights r, g, and b:

$$C = rR + gG + bB \tag{3.1}$$

The CIE RGB color-matching functions are shown in Figure 3.1. Achromatic light is represented by equal values of r, g, and b.

In any color space, it is also important to establish a precise definition for "white" light. The definition of white light is called the *reference white* for the color space. In the CIE RGB space, the reference white has tristimulus values $r=g=b=1.0$.

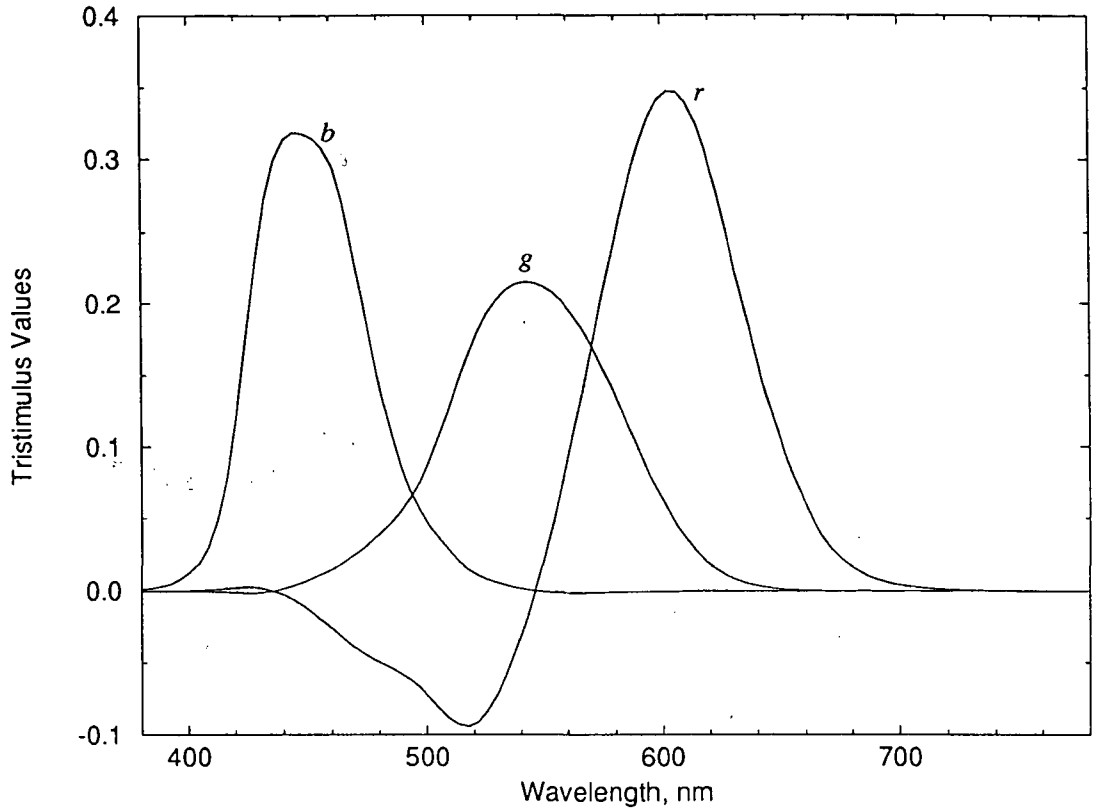


Figure 3.1: 1931 CIE RGB Color-Matching Functions

3.1.2 CIE XYZ Space

The CIE RGB space is useful in that it allows a standardized set of primary wavelengths to be used to reproduce all visible colors. However, it can be seen from Figure 3.1 that for some colors the matching coefficients take on negative values. This implies that an amount of the particular primary must be *added* to the color C to obtain a color match. To alleviate this problem, the CIE in 1931 introduced another color coordinate system whose color-matching functions have non-negative values, the XYZ space. The transformation between the CIE RGB and XYZ spaces is linear, and is given by:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.2}$$

The Y component of this color space indicates the strength of the achromatic portion of a light stimulus, also called the *luminance* of the source. The XYZ space is an often-used reference system, and the CIE has designated several standard illuminants for applications. The reference white for the CIE XYZ space is called the *equal-energy* white, which has tristimulus values $r=g=b=1.0$. The color-matching functions for the CIE XYZ color space are shown in Figure 3.2. A disadvantage of the XYZ system is that the three primaries no longer correspond to physical light sources.

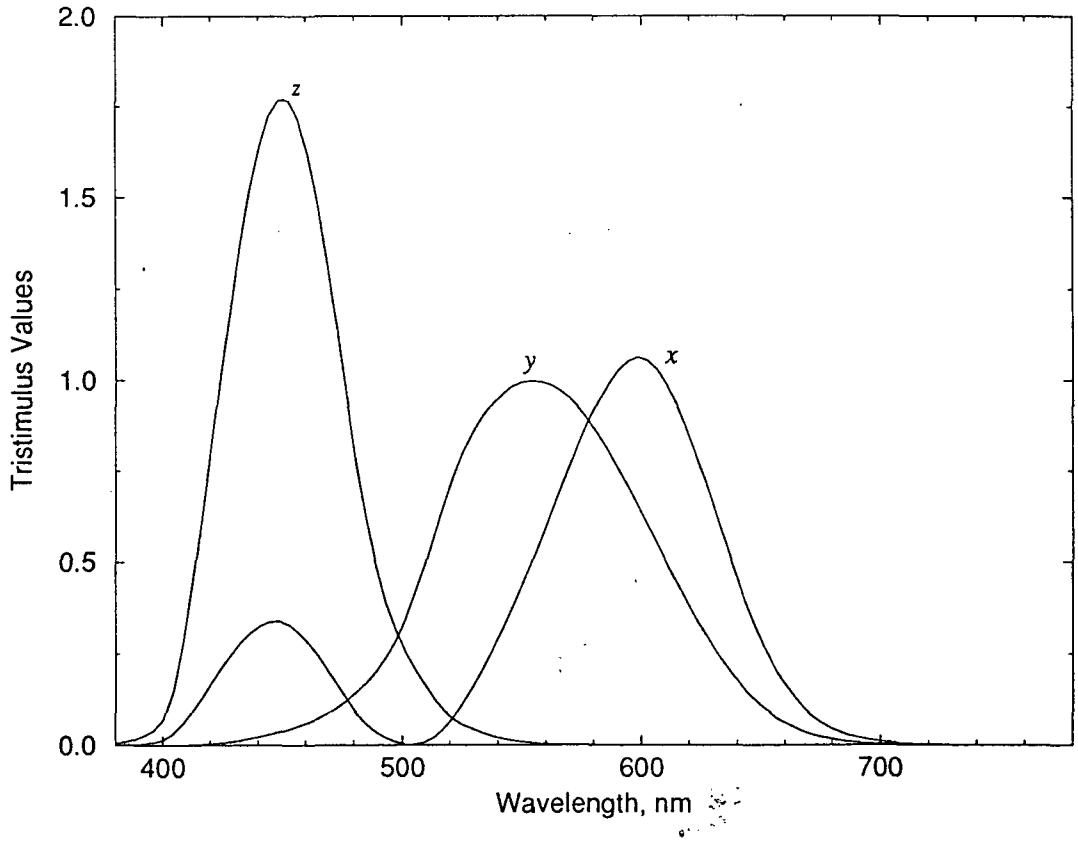


Figure 3.2: 1931 CIE XYZ Color-Matching Functions

3.1.3 NTSC RGB Space

While developing a standard for color television in North America, the National Television Standards Committee (NTSC), chose a color space with red, green, and blue primaries. These primaries correspond to the color properties of the phosphors used in television receiver CRTs, and so are known as the NTSC receiver primaries. These primaries differ slightly from those of the CIE. The transformation between the CIE XYZ space and the NTSC primaries is linear, given by:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.607 & 0.174 & 0.201 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.117 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.3}$$

As in the case of the CIE RGB primaries, the color-matching functions take on negative values for some colors. The reference white for this system was chosen to be CIE illuminant "C", for which the NTSC RGB tristimulus values are $r=g=b=1.0$. The NTSC RGB color primaries are commonly used as the basis for the representation of color digital images in most image display hardware.

3.1.4 NTSC YIQ Space

Along with their RGB system, the NTSC defined an additional color system to be used for transmission of the color television signal. This system was carefully constructed to allow backward compatibility with existing black-and-white receivers, while allowing the additional color information to be transmitted without requiring additional channel bandwidth. This system, known as the YIQ space, consists of an achromatic component Y (luminance), and two color-component signals I and Q (chrominance), which are quadrature modulated and added to the Y signal for transmission. Transformation between the NTSC RGB and YIQ spaces is linear:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.4}$$

3.2 Chromaticity

Returning to the CIE XYZ color space, a few additional comments are in order. The three coordinates X, Y, and Z allow a quantitative measure of all possible colors, at all possible luminance levels. For purposes of analyzing colors only, it is desirable to perform a normalization to remove the luminance component. If we choose the normalization:

$$x = \frac{X}{X+Y+Z}; \quad y = \frac{Y}{X+Y+Z}; \quad z = \frac{Z}{X+Y+Z} \tag{3.5}$$

we have a representation with only two independent quantities, since $x+y+z=1$. The quantities x, y, and z are called the *chromaticities* of the color. By plotting the chromaticities x vs. y, we obtain the CIE Chromaticity Diagram, shown in Figure 3.3. On this diagram, spectral (monochromatic) colors appear on the horseshoe-like outer edge of the figure. The remaining edge of the boundary is a straight line connecting the blue and red spectral colors, and is known as the "line of purples". All colors map to points within this boundary. Also shown on the diagram are the three NTSC RGB primaries. The triangular area formed by these three points corresponds to colors that may be represented by positively weighted linear combinations of these primaries. Thus we see that a system based on the NTSC primaries cannot represent all visible colors, although this is not a great limitation for most applications. Also shown on the diagram is the CIE standard illuminant "C", the NTSC reference white.

3.3 Perceptual Color Spaces

For lossy image coding, there are many instances where we desire a quantitative measure of system performance which indicates the amount of distortion introduced by the coding process.

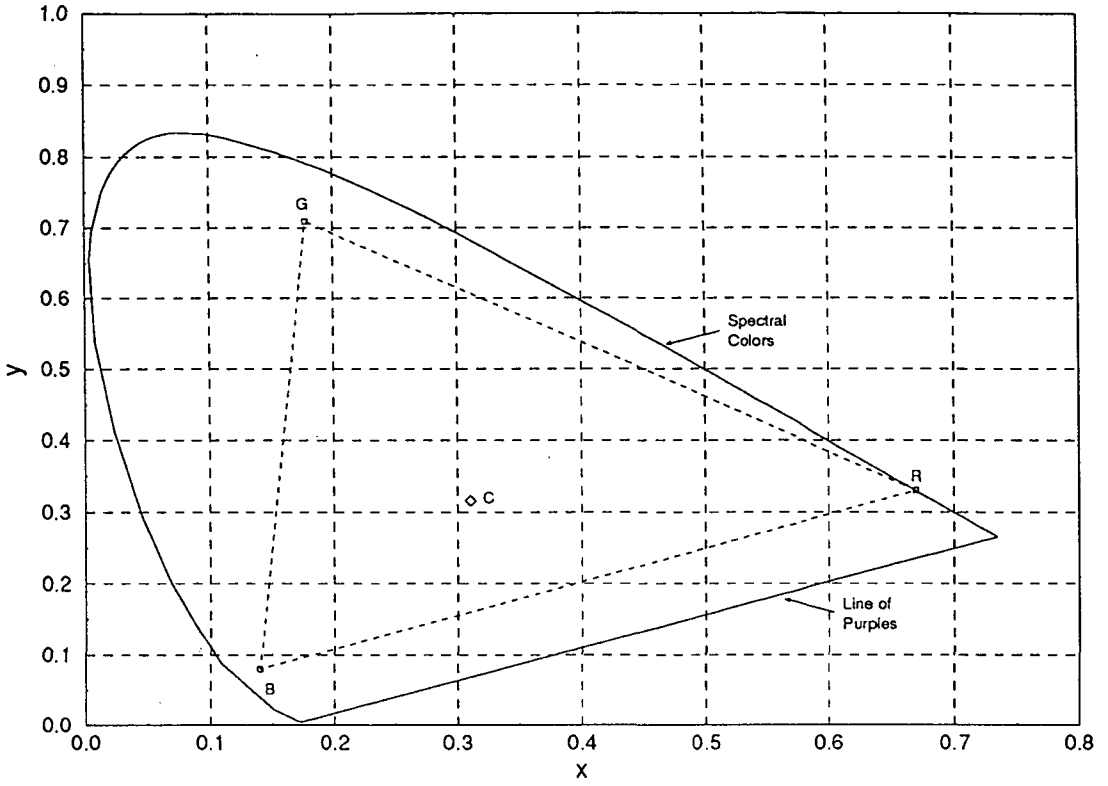


Figure 3.3: CIE Chromaticity Diagram

For achromatic images, a mean-squared error measure between the original image I and the reconstructed image $\hat{I}$ is usually involved, i.e.:

$$D = E\{(I_{ij} - \hat{I}_{ij})^2\} \tag{3.6}$$

where $E\{\cdot\}$ denotes expectation. In this measure, the quantity $(I_{ij} - \hat{I}_{ij})$ gives a measure of the "distance" between two scalar pixel values. For color images, we require a measure of the distance between two three-dimensional color vectors. Furthermore, since the goal of a progressive transmission is to present the image to a human observer, this distance measure should also be as consistent as possible with color differences perceived by a human observer. There are many experimental results in the literature for both achromatic and color stimuli which aid in defining this measure.

An important result of research into achromatic vision is the observation that the eye is most sensitive to relative, as opposed to absolute, changes in luminance values. This relative dependency is also known as luminance contrast. Contrast dependency is described by Weber's law [22], which states that if the luminance f_o of an object is just noticeably different from the luminance f_s of its surroundings, then their ratio is

$$\frac{|f_s - f_0|}{f_0} \approx \text{constant} \tag{3.7}$$

This effect is illustrated in Figure 3.4. As shown in the figure, Weber's law may be expressed in several ways, with logarithmic, square- and cube-root laws being most popular.

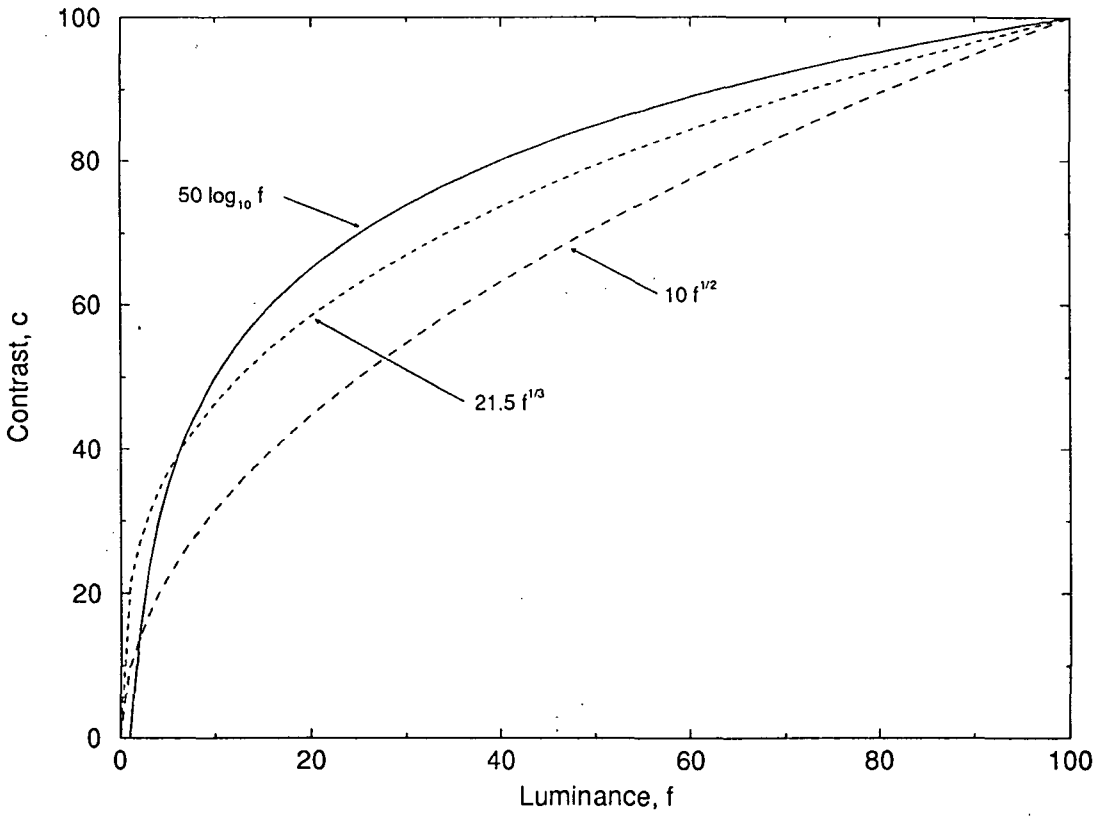


Figure 3.4: Weber's Law of Contrast Dependency

Important work considering the perceptual color differences was done in 1942 by D.L. MacAdam [23]. MacAdam's experiment involved over 25,000 trials to determine the just-noticeable color differences (JNDs) of color samples presented together to a single observer, Perley G. Nutting (PGN). The PGN observations and subsequent trials by MacAdam and others have yielded reasonably consistent JNDs. An example of these JNDs are shown on the CIE Chromaticity Diagram in Figure 3.5, enlarged by a factor of ten. In defining a color difference measure, we might wish to extend the formula for achromatic images, obtaining a simple Euclidean distance expression. However, the JNDs appear as ellipses on the diagram, indicating that a Euclidean distance measure is not sufficient to describe these differences. A constant-weighted Euclidean measure is also not appropriate, as the eccentricity of the ellipses varies throughout the diagram, indicating the need for non-constant weighting factors. It would be useful to find a color space in which JNDs appeared as circles (spheres in three-dimensions), allowing the use of the Euclidean distance measure.

3.3.1 CIE Uniform and Modified Uniform Chromaticity Scales

To satisfy the goals outlined in the previous paragraph, the CIE has made several attempts to define a color space for which a simple color difference formula exists. The first of these was introduced in 1960, called the Uniform Chromaticity Scale (UCS) [22]. The goal of this system is to transform the JND ellipses into circles, while maintaining a simple, invertible transformation to and from the CIE XYZ space. The transformation is given by

$$\begin{aligned} Y &= Y \\ u &= \frac{4X}{X+15Y+3Z}; \quad v = \frac{6Y}{X+15Y+3Z} \end{aligned} \tag{3.8}$$

where u and v are the chromaticity values for the UCS system.

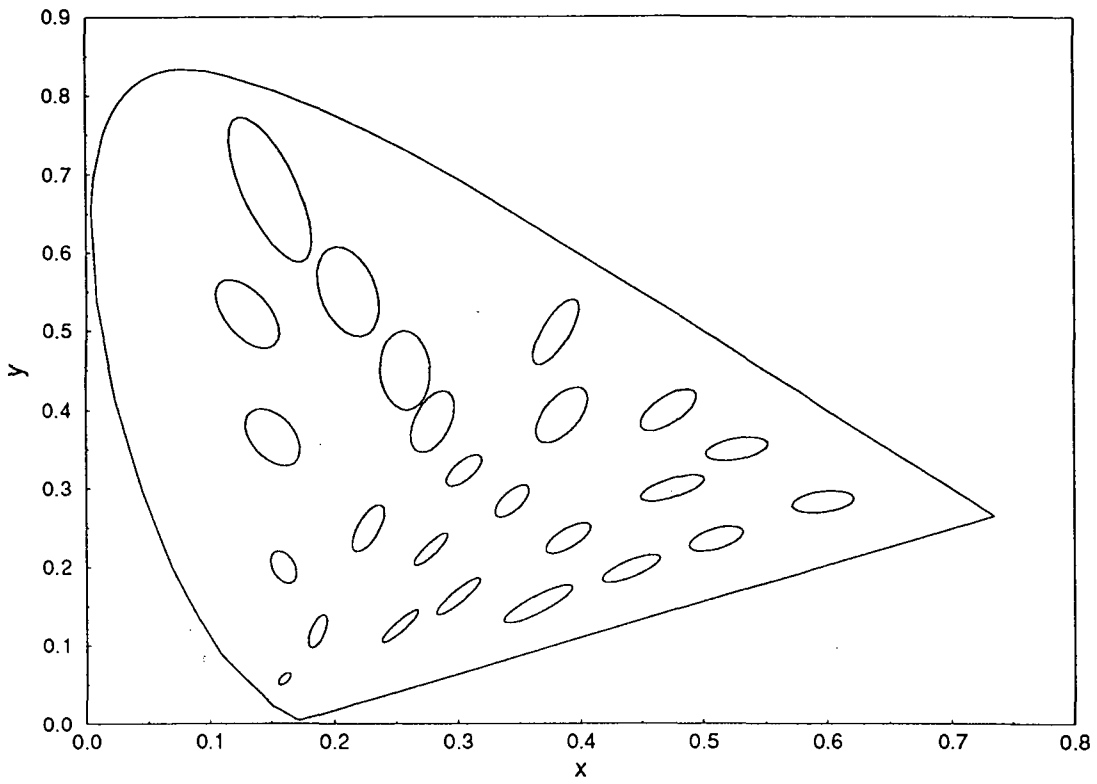


Figure 3.5: Just-Noticeable Color Differences

In 1964, the UCS was modified to include brightness effects, as described by Weber's law. The modified UCS system uses the tristimulus values U*V*W*, given by

$$\begin{aligned} W^* &= 25(100Y)^{1/3}-17, \quad 0.01 \le Y \le 1.00 \\ U^* &= 13W^*(u-u_0) \\ V^* &= 13W^*(v-v_0) \\ u_0, v_0 &= \text{chromaticities of reference white} \\ W^* &= \text{contrast or brightness} \end{aligned} \tag{3.9}$$

In this system, the difference between two colors, Δs, is defined as

$$(\Delta s)^2 = (\Delta U^*)^2+(\Delta V^*)^2+(\Delta W^*)^2 \tag{3.10}$$

3.3.2 CIE L*a*b* Perceptual Color Space

In 1976, the CIE defined two new color systems designed to give improved accuracy over the modified UCS system. The first, the L*a*b* space [24], is intended to be used for color differences on the order of the JNDs. It is defined as

$$
\begin{aligned}
L^* &= 25\left(\frac{100Y}{Y_0}\right)^{1/3} - 16, \quad 0.01 \le Y \le 1.00 \\
a^* &= 500\left[\left(\frac{X}{X_0}\right)^{1/3} - \left(\frac{Y}{Y_0}\right)^{1/3}\right] \\
b^* &= 200\left[\left(\frac{Y}{Y_0}\right)^{1/3} - \left(\frac{Z}{Z_0}\right)^{1/3}\right] \\
X_0, Y_0 Z_0 &= \text{reference white}
\end{aligned}
\tag{3.11}
$$

The quantity L^* is proportional to the perceived luminance, or brightness, of the light. The values a^* and b^* describe the red-green and yellow-blue content of the light, respectively. The color difference, Δs, is defined to be

$$
(\Delta s)^2 = (\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2 \tag{3.12}
$$

3.3.3 CIE L*u*v* Perceptual Color Space

The second color system introduced in 1976 by the CIE was the L*u*v* space [24], intended for use with larger color differences than the JNDs, on the order of those found in the Munsell *Book of Color*. It is defined as

$$
\begin{aligned}
L^* &= 25\left(\frac{100Y}{Y_0}\right)^{1/3} - 16 \\
u^* &= 13L^*(u'-u_0) \\
v^* &= 13L^*(v'-v_0) \\
u' &= \frac{4X}{X+15Y+3Z}; \quad v' = \frac{9Y}{X+15Y+3Z} \\
u_0 &= \frac{4X_0}{X_0+15Y_0+3Z_0}; \quad v_0 = \frac{9Y_0}{X_0+15Y_0+3Z_0}
\end{aligned}
\tag{3.13}
$$

and has the color difference formula

$$
(\Delta s)^2 = (\Delta L^*)^2+(\Delta u^*)^2+(\Delta v^*)^2 \tag{3.14}
$$

3.3.4 Other Perceptual Color Systems

The preceding color spaces use several known properties of color vision to allow more accurate estimates of perceived differences. Attempts have also been made to provide a model for the human visual system itself. In particular, we note two similar models by Frei [25] and

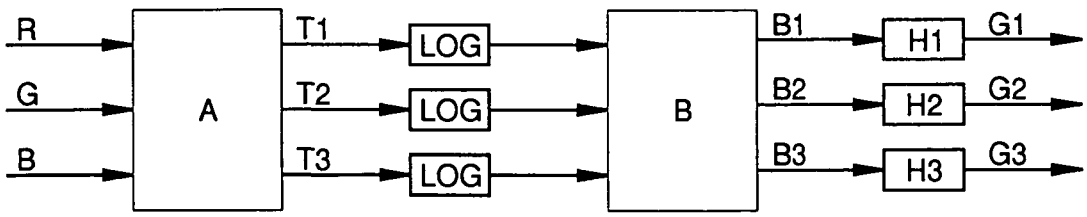


Figure 3.6: A Model For Color Vision

Faugeras [26]. A block diagram of these models is shown in Figure 3.6. The input red, green, and blue functions are linearly transformed, yielding outputs corresponding to the three types of color receptors (cones) in the human retina. A logarithmic transform is performed next, providing the effect described by Weber's law. A second linear transform is then performed, corresponding to neural processing of the cone outputs in the eye. Finally, the three signals pass through a spatial bandpass filter stage, modelled using experimental observations of the eye's response to spatial frequencies. Both authors indicate that a Euclidean distance measure corresponds well to perceived differences. They also suggest that this provides a suitable measure for assessing the quality of a coded image $\hat{I}$ from an original image I:

$$d(I,\hat{I}) = \sqrt{\sum_{i=1}^{3} \iint \left[G_i(x,y) - \hat{G}_i(x,y)\right]^2 dx\, dy} \tag{3.15}$$

where the integration is taken over the entire image.

Color vision models like the above can be useful in an image coding system, allowing the coder to "hide" errors from the viewer. In this thesis, perceptual color spaces are used to attempt to improve the quality of early progressive transmission passes. Also, the expression of (3.15) provides a meaningful, quantitative measure of the quality of the images produced by the coder. In this work, the model proposed by Frei is used for this purpose (see Section 5.2). To complete the Frei vision model, the values of the linear transformations A and B are given in (3.16).

$$A = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.607 & 0.174 & 0.201 \\ 0.000 & 0.066 & 1.117 \end{bmatrix}; \quad B = \begin{bmatrix} 21.5 & 0 & 0 \\ -41.0 & 41.0 & 0 \\ -6.27 & 0 & 6.27 \end{bmatrix} \tag{3.16}$$

3.4 Digital Image Display Hardware

To provide background and motivation for the remainder of the work in this thesis, we now examine some modern image display equipment. The introductory material in this section is taken from [27][28].

3.4.1 Achromatic Image Displays

Figure 3.7 illustrates a common configuration for the viewing of achromatic images, also called gray-scale images. To determine the intensity of the displayed pixel, the hardware retrieves an integer value (e.g., 8 bits) from the frame buffer memory, and uses a digital-to-analog converter (DAC) to convert the value to a voltage. The voltage is then passed to a CRT, where it is used to vary the strength of an electron beam. The beam strikes the phosphor on the face of the CRT, causing emission of light proportional to the strength of the beam.

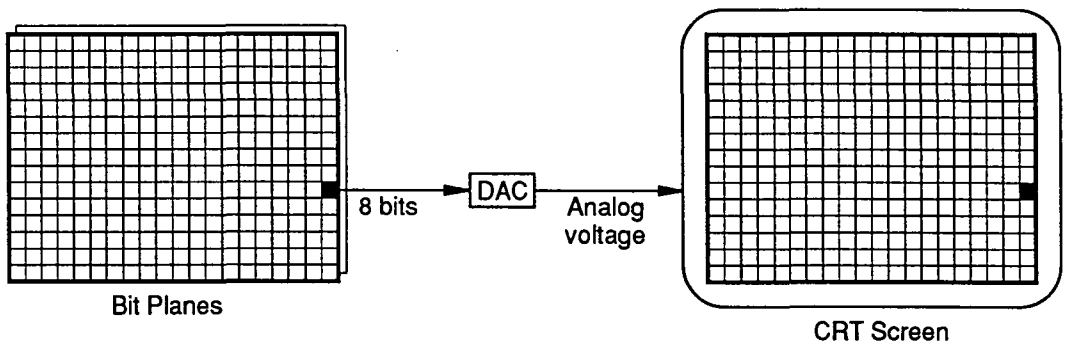


Figure 3.7: Achromatic Frame Buffer

The actual relationship between the voltage, V, presented to the CRT and the intensity, I, of the emitted light is a nonlinear one, given by

$$I = kV^{\gamma} \tag{3.17}$$

where k and γ are constants for the particular CRT. Typically, $\gamma \in [2.0, 2.5]$. Therefore, an additional nonlinear correction stage is often included in the display system, to eliminate the need

to "gamma-correct" the digital values. This was the case for the image display hardware used in this work, and so this issue was not considered further.

3.4.2 Full-Color Image Displays

The human eye can distinguish hundreds of thousands of different colors in a color space, depending on viewing conditions [27]. A full-color (also called true-color) frame buffer provides a means of displaying this wide range. Such a system is illustrated in Figure 3.8.

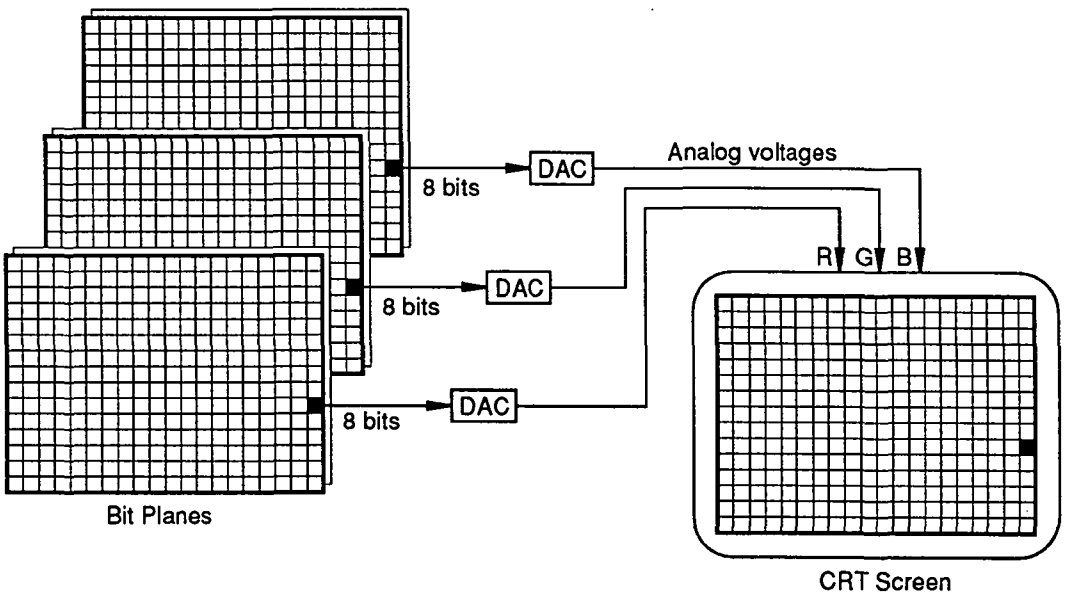


Figure 3.8: Full-Color Frame Buffer

The system shown uses an RGB color coordinate system, the most common choice. The red, green, and blue tristimulus values for a pixel are each represented by eight bits, allowing a total of 2^{24} colors to be displayed. More expensive systems may use more than 24 bits per pixel. The three 8-bit values are converted to voltages by the DACs, and passed to the CRT. The color CRT uses three electron beams to produce the displayed color. Each beam excites a particular type of phosphor in the CRT which emits light in either the red, green, or blue range. Referring to the previous discussion of the CIE XYZ space, recall that three visible primaries are not sufficient to

represent all visible colors. However, the full-color display offers excellent reproduction of most images. A disadvantage of the full-color display is the large amount of memory required to represent an image. This memory must be quickly accessible to allow real-time update of the CRT, making full-color image displays costly.

3.4.3 Pseudo-Color (Color-Mapped) Image Displays

Many applications of digital images benefit from, or require, color capabilities to be effective. If a full-color display is used, an application may become too costly to implement practically. Also, the images involved require large amounts of storage space, whether in display memory or on a mass-storage device. A less expensive solution is needed.

Reducing the number of bits per component in a full-color display may help, at the expense of the number of available colors. However, a full-color display system using as many as five bits per component (15 bits/pixel) exhibits noticeable contouring. In addition, it can be observed that most natural images do not make use of the full range of representable colors. Often a small subset of colors is all that is required to provide an acceptable representation to a human viewer. Computer graphics applications usually require a very small set of colors. Finally, some applications do not inherently include color, but may benefit from its use. For example, an X-ray image may be composed of 8-bit samples. By assigning specific colors to various sample ranges, areas of interest to the physician may be made more apparent.

These applications naturally lead to the pseudo-color or color-mapped frame buffer, shown in Figure 3.9. This type of display is typical of those found on personal computers and workstations. A smaller amount of image memory is required, one-third that of the full-color system example. The values stored in memory are used as indices into a 24-bit table, the colormap.

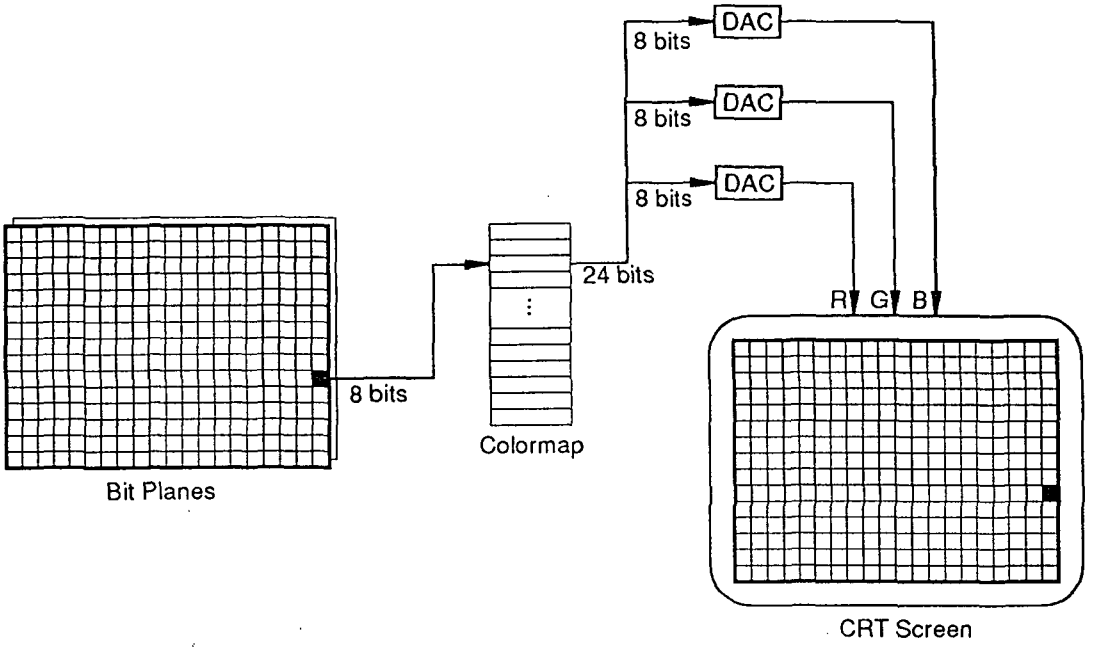


Figure 3.9: Color-Mapped Frame Buffer

Each entry in the colormap consists of 8-bit values for the red, green, and blue portions of the pixel. These three values are then passed through DACs to the red, green, and blue electron guns of the CRT, as with full-color system. The color-mapped system allows the display of a small number of colors at a time, 2^8 for the system shown in the figure, which can be selected from a larger set of colors (2^{24} for this example). By careful selection of the colors in the colormap, a large variety of images can be displayed, often with quality approaching that of a full-color display system. Issues concerning the design and use of colormaps is the subject of the next chapter.

4. Color Maps For Image Display

Chapter 3 introduced the pseudo-color image display system, and the concept of color-mapped images. The subject of creating a color-mapped image will now be addressed. In order to allow progressive transmission of these images, it will also be useful to examine some properties of the resulting color-mapped images.

The techniques for obtaining color-mapped images are application-dependent. For example, the choice of colors for a graphics application may be based solely on the preferences of the user. For other applications, a set of colors may be chosen which indicates critical areas in an intuitive manner to the user, perhaps using shades of red to indicate "hot" regions, and blues for "cold" ones. The following discussion is restricted to a particular type of color-mapped image, those that are derived from some form of full-color data in order to provide an acceptable representation to the user (using less data). It is a two-stage process: First select the entries of the colormap, then transform the original full-color data to use the colormap.

4.1 Selecting Colormap Entries

Consider the problem of selecting the entries in a colormap. The goal is to obtain a partitioning of a three-dimensional color space according to some rule, and then to select K representative points (the colormap entries) for each region in the partitioning. Several techniques for selecting colormaps available from the literature are discussed below, offering varying degrees of compromise between computational load and accuracy.

In each technique, the criteria used to select colors must be defined. Some of the techniques formulate color selection as an optimization problem, and select colors based on minimizing a cost function. Other techniques use more heuristic methods. All of the algorithms

perform the color selection assuming a color space with RGB-style primaries. Given the existence of perceptual color spaces, this would seem to be a poor choice. However, the use of an RGB space has several advantages from a computational standpoint. Most full-color image formats contain their data in an RGB format, usually using non-negative integer values to indicate the tristimulus values of the pixels. Therefore, use of an RGB space eliminates the need to perform nonlinear coordinate transformations. The integer RGB values are discrete and bounded, which can be used to reduce the memory requirements of a selection algorithm. Finally, it is observed that reasonable subjective results are still obtained in most cases. Computational efficiency is a concern for many of the techniques found in the literature, since one application of these techniques is to display full-color data on a pseudo-color display with minimal delay. The following techniques are given (approximately) in order of increasing computational load, which also tends to correspond to increasing subjective performance.

4.1.1 Popularity Algorithm

Heckbert [29] uses a simple technique for selecting the colors. Starting with a 24-bit full-color original, the image is first uniformly "pre-quantized" to 15 bits, five bits per component. This reduces the number of unique colors in the image to a more tractable number. From a histogram of these colors, the K entries in the colormap are chosen to be the K most frequent colors in the histogram. Although simple, Heckbert reports that this method yields poor subjective results, especially when the input image contains many different colors.

4.1.2 Median-Cut Algorithm

In a second technique, Heckbert [29] again pre-quantizes the 24-bit original image to 15 bits. Using the histogram, the K colormap entries are chosen so that each entry in the colormap occurs approximately an equal number of times in the resulting image. The algorithm recursively divides RGB space, starting from a single three-dimensional "box" containing all of the 15-bit

colors of the image. At each step, a box is split into two new boxes by passing a plane through it at some point. To simplify the algorithm and maintain the box-like partitions, the cutting planes are always constrained to be normal to one of the RGB coordinate axes. The method proceeds as follows:

1. "Shrink" the current box dimensions so that it just encloses the points.
2. Sort the points in the box along its longest axis (red, green, or blue).
3. Split the box into two boxes at the median point on that axis.
4. Continue splitting until K colors are obtained.

During the box-splitting process, the distribution of the points in the box may be such that the box cannot be split. That is, splitting the box would cause one of the two resulting boxes to contain no points. In this case, Heckbert suggests that these empty boxes can be used to further subdivide larger boxes. The median-cut algorithm is observed to perform better than the popularity algorithm, but may still exhibit poor performance (see next section).

4.1.3 Variance-Based Color Quantization

Wan, et al. [30], present a data clustering algorithm based on the minimization of a sum-squared error measure. The colormap entries are selected as the centroids of clusters formed by the input data vectors. They observe that Heckbert's median-cut algorithm has very little theoretical basis for use in selecting a distortion-optimal set of colors, and so can result in large color errors. They suggest that a more optimal method of partitioning is to subdivide the space in a way that minimizes a probability-weighted variance of the resulting clusters.

The color selection problem is defined as follows. Given a set of N points $\{s_i\}$ in three-dimensional space, select a smaller set of K representative points $\{r_j\}$ which provide a good representation in some sense. For an image, K might be 2^8, and N might be on the order of 2^{15} or 2^{24}, depending on the image. Since most full-color images contain fewer than 2^{24} pixels, the actual number of unique image colors will typically fall somewhere between these two values, but

may be even less than 2^{15}. The performance measure for selecting the representatives is the mean-square Euclidean distance between the input and representative points:

$$D = E\{\|s_i - r_j\|^2\} \tag{4.1}$$

where r_j is the representative point closest to the input point s_i in a Euclidean distance sense. Therefore, the problem becomes one of optimization, to select $\{r_j\}$ to minimize the distortion function D. However, the authors point out that finding a global minimum solution for the distortion function is an *NP*-complete problem, and its large size makes it computationally intractable to solve exactly. They then present an efficient, non-iterative technique for obtaining a good-quality sub-optimal solution.

Like the median-cut algorithm, the variance-based method begins with a single three-dimensional box in an RGB-based color space, containing all of the colors in the image. To reduce storage requirements, the color vectors can be pre-quantized to 15 bits (5 bits/component). At each iteration of the algorithm a box is chosen to be divided. As with the median-cut algorithm, the partitioning of the color space is constrained to rectangular boxes, with dividing planes which are normal to one of the coordinate axes. The place at which the particular axis and the plane intersect is called the *optimal cut-point.* The representative point for each box is chosen as the centroid of the box. The rules for choosing the box to divide and the optimal cut-point are presented below.

The authors first define a *weighted error variance* measure to describe the contribution of a particular cluster to the total distortion (i.e., the quantization error):

$$\delta_l^2 = W_l \sigma_l^2 \tag{4.2}$$

where σ^2 is the variance of error between the data points in cluster l and the representative point r_l for that cluster, and W_l is the fraction of the total number of data points contained in that

cluster. Thus, W_l reflects the probability that cluster l will be used. At each iteration, the box chosen to be divided is the one with the largest weighted variance.

After the box has been chosen, the optimal cut-point must be determined. The box will be split by a plane normal to of the coordinate axes. Therefore, three "axis-optimal" cut-points will be computed, one for each axis, and the best one will be selected as the optimal cut-point for the box. If a box, Ω_l, is split into two new boxes Ω_{l1} and Ω_{l2}, the quantization error contributed by the two boxes is the weighted sum of the variances of the two new boxes:

$$E = W_{l1}\sigma_{l1}^2 + W_{l2}\sigma_{l2}^2 = \hat{\sigma}_{l1}^2 + \hat{\sigma}_{l2}^2 \tag{4.3}$$

Therefore, the optimal cut-point for a particular axis is the cut-point which minimizes the sum of the weighted variances (quantization error) of the two newly-formed boxes.

The search for the optimal cut-point can also be time-consuming, so the authors make an additional simplification: Project the three-dimensional data points onto the particular axis under consideration, reducing the search to one-dimensional quantities. The complete algorithm is summarized below:

1. Select the box with the largest weighted variance for dividing.
2. Project all data points in the box onto each of the three coordinate axes. For each of the three projected distributions, calculate the optimal cut-point and the reduction of expected variance.
3. Partition the box using the plane perpendicular to the axis along which the reduction of expected variance is the largest. This plane intersects the axis at the optimal cut-point.
4. Compute the weighted variance for each of the two new boxes.
5. Repeat steps 1 through 4 until K boxes are formed.
6. Calculate the centroids of the resulting boxes. These will be the K entries of the colormap.

The authors report good performance of the algorithm in terms of required storage and execution speed. Furthermore, the obtained colormaps have a lower mean-squared error than the median-cut algorithm, and come very close to the performance obtained by the k-means clustering algorithm, which is known to locate local minima (see next section).

4.1.4 Iterative Clustering Algorithms

Linde, Buzo, and Gray [18] proposed a clustering algorithm for use in the design of vector quantizers. This algorithm is nearly identical to the iterative k-means algorithm [31] used in cluster analysis. The formulation of the problem is the same as that described in the previous section. The approach is to use a set of N representative data vectors to iteratively locate K cluster centers (centroids), while minimizing a distortion measure such as mean-squared error. Initially, the K cluster centroids may be selected as K input data vectors, or they may be iteratively constructed by perturbing a single vector. The algorithm is terminated when the change in the distortion measure between successive passes falls below a user-determined threshold. Constraints on the shape of the partitions imposed by the other techniques in this section (i.e., rectangular boxes) are not present, making it possible to achieve a lower distortion level. Such an algorithm is known to converge to a local minimum under certain conditions [18][32]. However, its iterative nature results in long computation times to converge to a minimum cost, and is somewhat sensitive to the initial conditions of the algorithm.

4.2 Converting The Image To Use The Colormap

After the colormap is chosen, the original color image must be converted to use the colormap. This is simply the application of a three-dimensional vector quantizer, using the colormap as the quantizer codebook. Hence, the process of creating a color-mapped image from a full-color one is usually known as *color quantization*. Each entry in the colormap is assigned an index number for future reference. The nearest (in a Euclidean distance sense) entry to the original color vector s_i is located in the codebook (colormap). The index of the colormap entry is then placed in the output image. Some of the techniques discussed above also provide techniques for quickly locating the nearest colormap entry. Thus, the original $P \times Q$ full-color image is reduced to a K-entry colormap, and a $P \times Q$ array containing the $(\log_2 K)$-bit index values

for each pixel.

4.3 Source Images For This Work

Four full-color source images were chosen for use in construction and evaluation of progressive transmission schemes for color-mapped images. Unless otherwise noted, the image is 512×512 pixels, 24-bits per pixel (8 bits per red, green, and blue components). They are:

- **Hat:** A head-and-shoulders view of a woman wearing a hat
- **Park:** A view of a lake surrounded by trees on a sunny day
- **Omaha:** A satellite view of a city as might be available from a geographic database. The image was taken by the Thematic Mapping System (TMS). The image was formed from TMS emission wavelength bands 2 (visible green), 3 (visible red), and 4 (near infrared). The image was padded from 477 to 512 columns using black pixels.
- **Lincoln:** A satellite view of a city taken by the France's SPOT satellite sensor. The image was formed from the three SPOT bands, which record emissions in the visible green, visible red, and near infrared ranges.

From these images, color-mapped versions were obtained by selecting an 8-bit (256-entry) colormap and quantizing the 24-bit image to the colormap. Selection of the colormap entries was done using the variance-based algorithm of Wan, et al., described in Section 4.1.3. It was felt that this algorithm provided a good tradeoff between the resultant image quality and execution speed, and would provide images typical of what would be encountered in an application (e.g., a geographical database). Several experiments were also performed using Heckbert's median-cut method to select colors, but subjective tests showed that the resultant images often contained obvious color artifacts, whereas the variance-based method yielded a much more graceful degradation in image quality. The 24-bit original images and the resulting 8-bit color-mapped images are shown in Figures 4.1 through 4.8 at the end of this chapter.

4.4 Properties of Color-Mapped Images

Since a source coder will be developed for use in progressive transmission, it will be useful to study the properties of color-mapped images. As will be demonstrated, color-mapped

images present several problems from a coding standpoint, especially when used in a progressive transmission system. This places restrictions on the type of coding which can be performed.

In a standard image coding scenario, there are several known properties of images which can be exploited to obtain data compression. Perhaps the most useful trait of image data is that pixel-to-pixel correlation nearly always exists in the data, especially if the image is a natural scene. For an achromatic image, this means that the integer pixel values (which describe the intensities of the pixels) will be numerically similar for spatially adjacent pixels. For a full-color image, a similar condition exists for adjacent pixels on individual color planes. This correlation is what allows predictive coding schemes (e.g., DPCM) to perform efficient compression. In a color-mapped image, the values stored in the pixel array are no longer directly related to the pixel intensity (or the magnitude of one of the color components). Two color indices which are numerically adjacent (close) may point to two very different colors. Hence, the correlation between pixels appears to be lost. It does, of course, still exist, but only via the colormap. This fact can be exploited, and will be addressed later in this chapter.

Another measure which is used to describe an image is its entropy. Entropy provides a measure of the randomness of a source, based on an assumed model for that source. Treating the image as a memoryless source with an alphabet S containing R symbols, the zeroth-order entropy, H_0, is defined as

$$H_0 = -\sum_{i=1}^{R} P(S_i) \log_2 P(S_i) \quad \text{bits} \tag{4.4}$$

where $P(S_i)$ is the probability of occurrence of symbol S_i. If there is correlation between adjacent pixels, another possibility is to consider a first-order model for the image. If the image is transmitted as a one-dimensional source in a row-by-row (or column-by-column) manner, a first-order differential entropy, H_1, can be defined on an alphabet D consisting of the $2R$-1 possible

differences between the elements of alphabet S:

$$H_1 = -\sum_{j=1}^{2R-1} P(D_j) \log_2 P(D_j) \quad \text{bits} \tag{4.5}$$

These quantities were computed using the index arrays for the four test images, and are shown in Table 4.1.

Table 4.1: Entropies of the Source Images

Image	H_0	H_1
Hat	7.617	7.413
Park	7.470	7.797
Omaha	7.242	7.165
Lincoln	5.916	6.674

The large values of H_1 in the table verify that the spatial correlation in the image pixel values has been reduced by the color-mapping process. The values of H_0 are also relatively large, a direct result of selecting an 8-bit colormap. The data compression due to the color quantization process implies that the color index values stored in the image are more critical than similar values in, for example, an achromatic image. To further verify this, an experiment was conducted in which errors were introduced in the least significant bit of a color-mapped image, similar to what might be encountered if the color index values were quantized. The resultant images were of poor subjective quality at best, and often completely unrecognizable. Since quantization is a part of many popular source coding schemes, the available choices become limited.

When transmitting a $P \times Q$ color-mapped image, most of the data to be sent consists of the PQ color index values. However, it has been shown that much of the structure of the image has been disguised by the colormap. If the entire frame is to be coded losslessly, it is still possible to devise a scheme for transmitting the image which also achieves compression [33]. Progressive

transmission, however, requires that several *lossy* representations of the image can be obtained. In addition, these lossy versions must be of some use to the user, and so must convey some information about the content of the image. These requirements place severe restrictions on how the image is coded. It seems reasonable, however, that if some type of organization could be given to the colormap, the correlation between adjacent pixel values (colormap indices) could be restored, allowing the image to be coded more easily. This idea of "colormap sorting" is discussed in Section 4.5.

4.5 Colormap Sorting

In Section 4.4, several problems unique to color-mapped images were exposed. The root of these problems is that the colormap indices stored in the image have little relationship with each other, which complicates coding for progressive transmission. In this section, methods of restoring this relationship are discussed.

Sorting the colormap can be done to satisfy one of two possible goals. The first is the desire to restore correlation among the pixels to allow them to be efficiently coded, i.e., entropy reduction. The second goal is to allow the introduction of small errors in the color index values, such as those resulting from quantization, without a large reduction in subjective image quality. Even in this latter case, the desire for entropy reduction is implied since that is the purpose of quantization. These two goals conflict somewhat since the eye's sensitivity to color errors is dependent on many things, as was discussed in Chapter 3. It will be useful to find a solution which satisfies both of these goals to some degree.

Colormap sorting is a combinatorial optimization problem. Treating the K colormap entries as vectors, the problem is defined as follows. Given a set of vectors $\{a_1, a_2, ..., a_K\}$ in a three-dimensional vector space and a distance measure $d(i,j)$ defined between any two vectors a_i and a_j, find an ordering function $L(k)$ which minimizes the total distance D:

$$D = \sum_{k=1}^{K-1} d\big(L(k),\ L(k+1)\big) \tag{4.6}$$

The ordering function L is constrained to be a permutation of the sequence of integers $\{1, ..., K\}$. Another possibility results when the list of colormap entries is considered as a ring structure. That is, the colormap entry specified by $L(K)$ is now considered to be adjacent to the entry specified by $L(1)$. In this case, an additional term of $d(L(K),L(1))$ is added to the distance formula D.

The sorting problem is similar to the well-known travelling salesman problem, and is identical if the colormap is considered as a ring structure. As such, the problem is known to be *NP*-complete [34], and the number of possible orderings to consider is ½[(*K*-1)!] [35]. Algorithms exist which can solve the problem exactly [35][36]; however, these algorithms are computationally feasible only for K no greater than about 20. Efficient algorithms for locating a local minimum exist [35] for $K \leq 145$. For large colormaps such as $K=256$, another approach is necessary. Two techniques were tested. The first is a "greedy" technique, discussed in Section 4.5.1. The second is an algorithm which has performed well in practice, known as *simulated annealing*. Simulated annealing was chosen as the sorting method for the colormaps in this work†, and is described in more detail in Section 4.5.2.

To complete the problem definition above, the distance metric d must be determined. As discussed in Chapter 3, there are several possibilities depending on the color space used. For this work the distance metric was chosen to be (unweighted) Euclidean distance, and different color spaces were investigated. Three color spaces were selected: the NTSC RGB space, the CIE L*a*b* space, and the CIE L*u*v* space. The NTSC RGB space was chosen since it

†Late in the course of this research, the author was made aware of other optimizing techniques for this type of problem, using results from the field of graph theory [47]. Like simulated annealing, these methods produce suboptimal solutions. Unlike simulated annealing, however, they also offer known bounds on the optimality of the solution they provide. They also seem to offer a considerable speed advantage over simulated annealing. Due to time constraints, these methods were not investigated further, but they promise improvement for the colormap sorting process.

corresponds to the color primaries of the original images. Color spaces which can be linearly transformed to the NTSC RGB space were not considered, since the use of an unweighted Euclidean distance measure would give similar results for such a color space. The two CIE color spaces were selected since they provide a means to measure perceptual color differences.

4.5.1 Greedy Sorting Algorithm

The first colormap sorting algorithm investigated was a greedy algorithm. As the name implies, this is simply a "take what you can get" approach. The algorithm begins by selecting a starting node (i.e., a color vector). From this node, proceed to a node which has not yet been visited by selecting the path with the least cost. For the colormap sorting problem, the "path" is the distance between the colors, the function $d(i, j)$ defined in Equation 4.6. The algorithm proceeds until all nodes (colors) have been visited. To avoid any penalties due to the choice of the starting node, a path was formed starting at each of the 256 nodes. From the 256 paths, the path of least cost was then selected.

Tests using the greedy sorting algorithm indicated that it was not very successful in sorting the colormaps. The resultant images were still quite sensitive to small errors in the color indices, and had high differential entropies. The simulated annealing algorithm in the next section was able to provide better results in both categories.

4.5.2 Sorting Using Simulated Annealing

Simulated annealing [34][37] is a stochastic technique for combinatorial minimization. The basis for the technique comes from thermodynamics and observations concerning the properties of materials as they are cooled. The technique described in this section is based on the implementation in [37].

To illustrate this concept, consider an iron block. At high temperatures, the iron molecules move freely with respect to each other. If the block is *quenched* (cooled very quickly),

the molecules will be locked together in a high-energy state. On the other hand, if the block is *annealed* (cooled very slowly), the molecules will tend to redistribute themselves as they lose energy, with the result being a lower energy lattice which is much stronger. The distribution of molecular energies is characterized by the Boltzmann distribution:

$$P(E) \sim e^{-E/kT} \tag{4.7}$$

where E is the energy state, T is the temperature, and k is Boltzmann's constant. The significance of this distribution is that even at low temperatures, there is some probability that a molecule will have a high energy. In a combinatorial optimization situation, the Boltzmann distribution can be used to temporarily allow increases in the cost function, while still generally striving to achieve a minimum.

In the travelling salesman problem, the goal is to visit each city exactly once and return to original city with minimum path cost. Similarly, solving the colormap sorting problem involves selecting each color only once while minimizing the sum of the distances between the colors. To find a solution using simulated annealing, an initial path through the nodes (cities, colors) is chosen, and its cost computed. The algorithm then proceeds as follows:

1. Select an initial temperature T and a cooling factor α.
2. Choose a temporary new path by perturbing the current path (see below), and compute the change in path cost, $\Delta E = E_{new} - E_{old}$. If $\Delta E \leq 0$, accept the new path.
3. If $\Delta E > 0$, randomly decide whether or not to accept the path. Generate a random number r from a uniform distribution in the range [0, 1), and accept the new path if $r < \exp(-\Delta E/T)$.
4. Continue to perturb the path at the current temperature for I iterations. Then, "cool" the system by the cooling factor: $T_{new} = \alpha T_{old}$. Continue iterating using the new temperature.
5. Terminate the algorithm when no path changes are accepted at a particular temperature.

The decision-making process is known as the *Metropolis algorithm.* Note that the decision process will allow some changes to the path which increase its cost. This makes it possible for

the simulated annealing method to avoid easily being trapped in a local minimum of the cost function. Hence, the algorithm is less sensitive to the initial path choice. Reference [34] shows that if certain conditions are satisfied, the simulated annealing technique can asymptotically converge to a global optimum. Even in cases where it does not converge to the optimum, the method often provides high-quality solutions.

In the above description, initial values for T, α, and I need to be selected. Selection of these values requires some experimentation, although a few guidelines are provided in the references. For the images of this work, initial values of T ranged from 80 to 500, depending on the color space used. The cooling factor α was usually chosen as 0.9. The simulated annealing algorithm seemed to be most sensitive to the choice of this value, as values outside the range [0.85, 0.95] caused the cooling to occur too slowly or too quickly. The number of iterations per temperature I was chosen as 100 times the number of nodes (colors), or 25,600. However, to improve the execution speed of the algorithm an improvement suggested by [37] was added, which causes the algorithm to proceed to the next temperature if (10)(number of nodes) = 2560 successful path changes are made at a given temperature.

Also, a method for perturbing the path must be selected. In this work, the perturbations were made using the suggestions of Lin [35][37]. At each iteration, one of two possible changes to the path are made, chosen at random. The first is a *path transport*, which removes a segment of the current path and reinserts it at another point in the path. The location of the segment, its length, and the new insertion point are chosen at random. The second perturbation method, called *path reversal*, removes a segment of the current path and reinserts it at the same point in the path, but with the nodes in reverse order. The location and length of the segment are again randomly chosen.

The algorithm outlined in the previous paragraphs formulates colormap sorting as a

travelling salesman problem. This type of problem usually assumes a complete tour will be made (i.e., the salesman desires to return to the original city). Hence, the colormap is assumed to have a ring-like structure. However, the simulated annealing technique can also be used if this is not the case, allowing the colormap to be considered as a linear list structure. Experiments using both structures were conducted.

The results of sorting the colormaps of the test images using simulated annealing are shown in the following tables. Table 4.2 shows results for sorting the colormap as a circular ring structure, while Table 4.3 shows the results of sorting the colormap as a linear structure. Given in the tables are values for the resulting first-order entropy and the final path cost (the distance measure D).

Table 4.2: Resultant Images With Circularly Sorted Colormaps

Image Name	RGB Space		L*a*b* Space		L*u*v* Space	
	Cost	H_1	Cost	H_1	Cost	H_1
Hat	13.88	5.641	857.80	5.627	208.49	5.480
Park	19.32	6.325	1609.46	6.330	310.41	6.218
Omaha	11.04	6.209	1081.82	6.303	363.21	6.178
Lincoln	10.62	5.513	1193.88	5.831	224.06	5.478

Table 4.3: Resultant Images With Linearly Sorted Colormaps

Image Name	RGB Space		L*a*b* Space		L*u*v* Space	
	Cost	H_1	Cost	H_1	Cost	H_1
Hat	11.68	5.575	847.29	5.933	200.31	5.512
Park	15.66	6.260	1509.25	6.775	292.29	6.546
Omaha	10.81	6.532	1004.69	6.554	283.66	6.199
Lincoln	10.61	5.774	1177.80	6.120	204.64	5.735

Note that the zero-order entropy H_0 is not changed by the sorting process, since permuting the colormap entries does not change the frequency of occurrence of a particular color. The lower first-order entropies of the resultant images indicate that some of the spatial correlation between color indices has been restored in each case. The sorting results for the NTSC RGB space show that sorting in this space yields good results, if entropy reduction (the first goal stated above) is the goal. However, the L*u*v* space sorting gives better results, with the added advantage that the perceptual differences between colormap entries has been considered. Hence, the resultant images from this sort should also be able to accept quantization errors while maintaining good subjective quality, the second goal stated previously. To verify this hypothesis, the quantization experiment of Section 4.4 was repeated with the L*u*v*-sorted images. Good subjective results were obtained using quantization levels down to as low as 5 bits/pixel from the 8-bit original. Figure 4.9 shows the colormap for the Park image, before and after sorting. The sorted colormap shown was sorted as a linear list in L*u*v* space. Figure 4.10 shows the result of quantizing the Park image to 5 bits/pixel, before and after the colormap has been sorted.

In this chapter, colormap design techniques were discussed. It was also shown that some of the problems associated with coding a color-mapped image can be overcome to a degree. In the next chapter, these ideas are used to complete a progressive transmission scheme for color-mapped images.

Figure 4.1: Hat, Full-Color Original

Figure 4.2: Hat, 8-bit Color Map

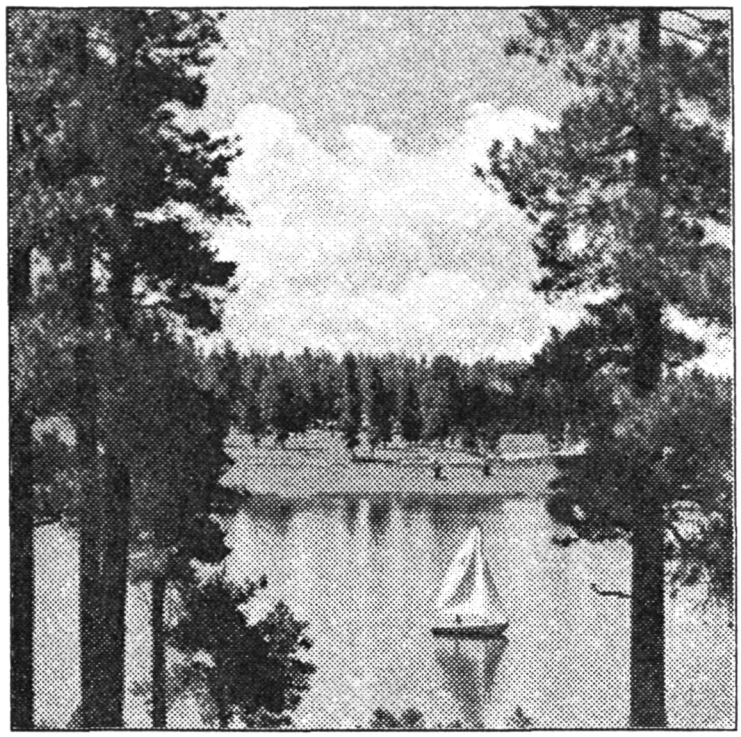


Figure 4.3: Park, Full-Color Original

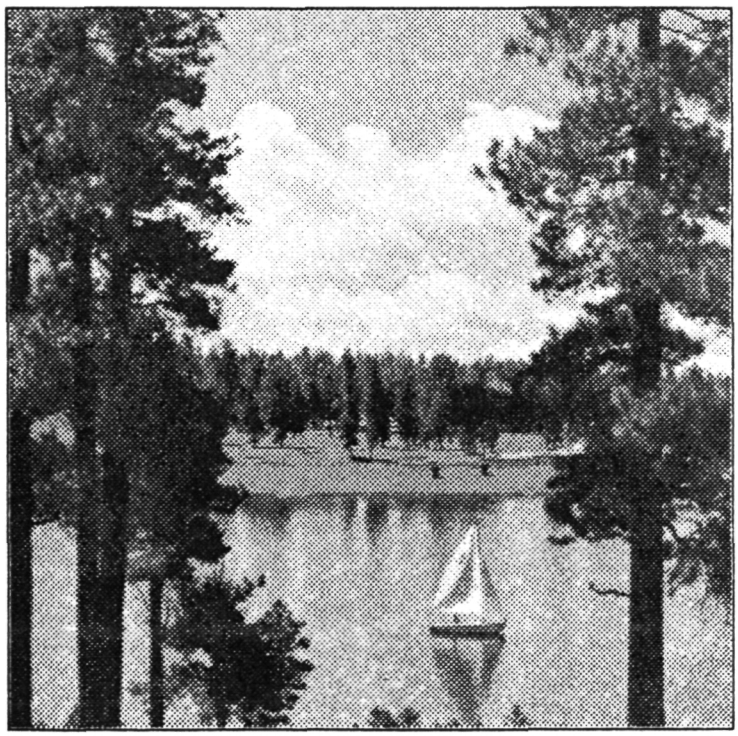


Figure 4.4: Park, 8-bit Color Map

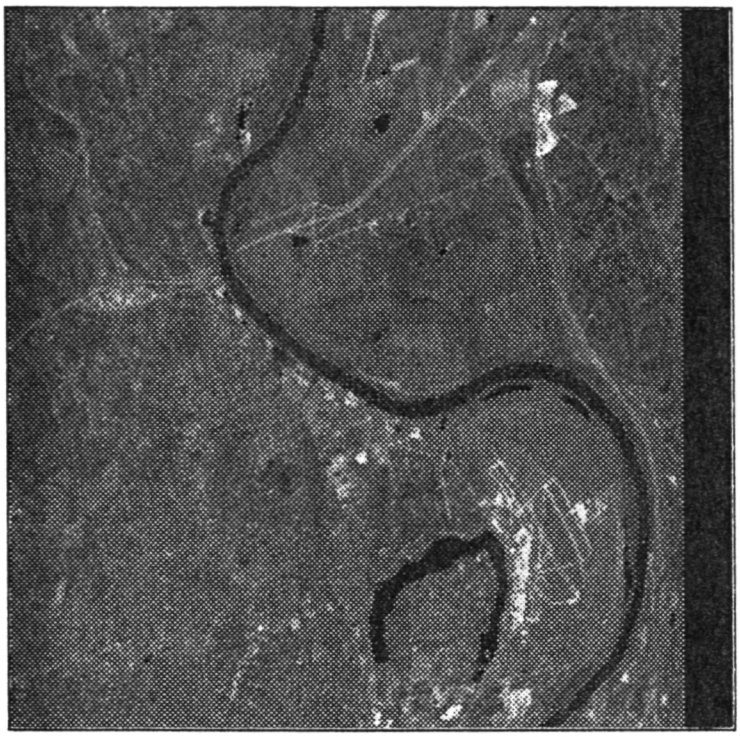


Figure 4.5: Omaha, Full-Color Original

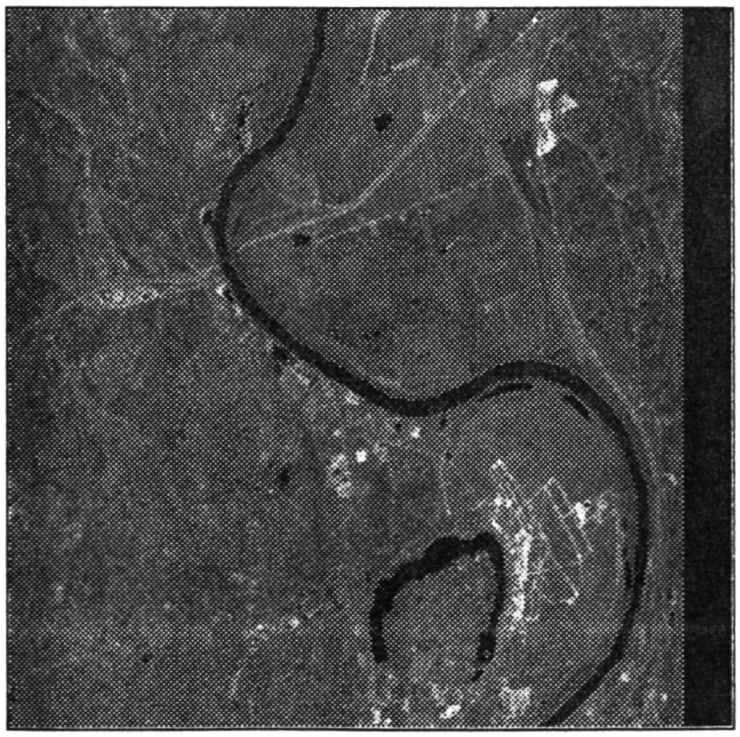


Figure 4.6: Omaha, 8-bit Color Map

Figure 4.7: Lincoln, Full-Color Original

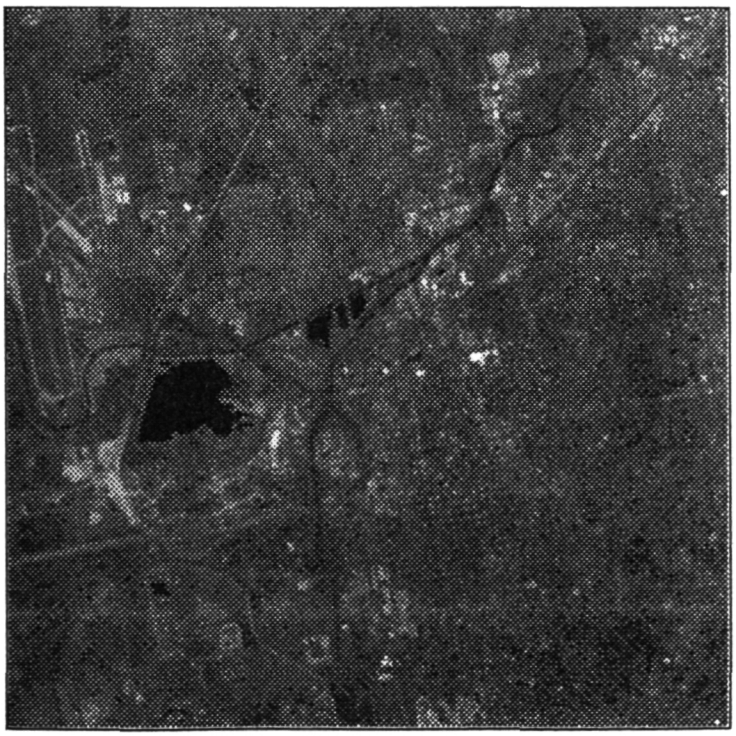
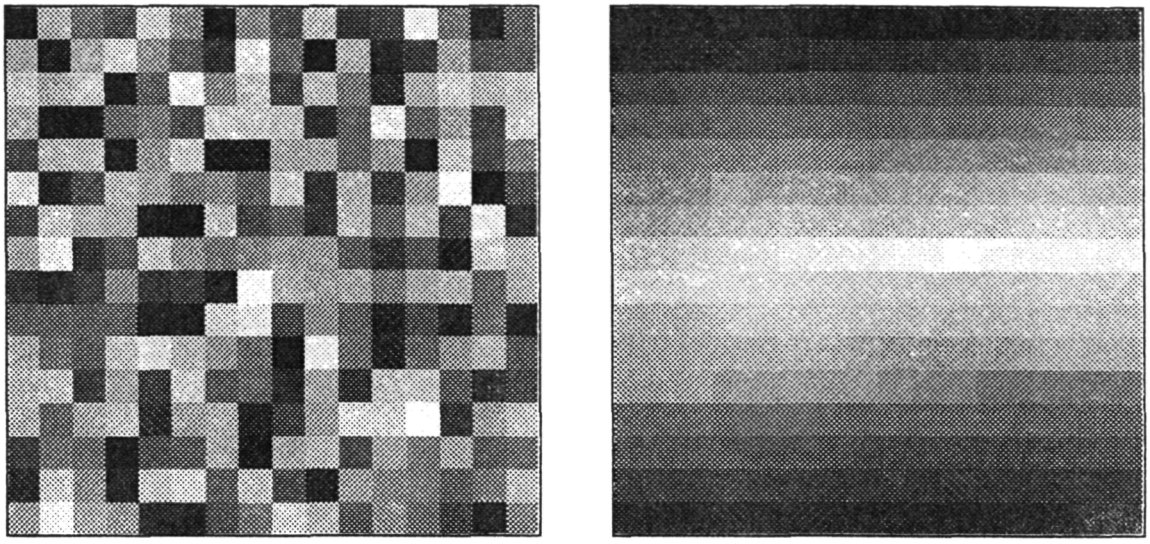
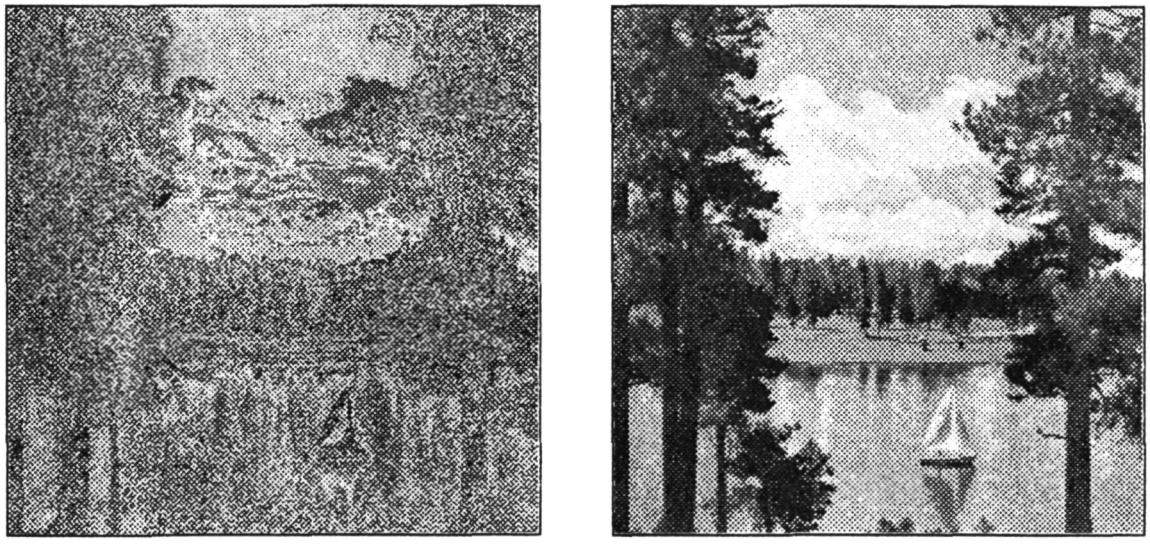


Figure 4.8: Lincoln, 8-bit Color Map



Original Color Map Sorted Color Map

Figure 4.9: Original and Sorted Color Maps, Park Image



Original Color Map Sorted Color Map

Figure 4.10: Park, Original and Sorted Color Maps, 5 bits/pixel

5. A PT Scheme For Color-Mapped Images

In this chapter, the ideas developed in previous chapters are combined to realize a progressive transmission (PT) scheme for color-mapped images. The results of simulating this scheme are evaluated, and various modifications to improve performance are investigated.

5.1 A Basic CMPT Framework

The first step toward obtaining a color-mapped progressive transmission (CMPT) system is to determine a suitable general framework for transmitting the image data. As discussed in Chapter 2, the possibilities may be split into the two categories of transform vs. non-transform schemes. To investigate transform-type schemes, the discrete cosine transform (DCT) was performed on the test images at different coding rates. The coder used had the same structure as the PIT coder of Chitprasert and Rao [10] (see Chapter 2), except that the HVS coefficient weighting function of their scheme was removed. The subjective quality of the resulting images was extremely poor, except at very high rates (5 bits/pixel or more). Upon further investigation, it was observed that although a DCT coder captures most of the energy of the image, it is possible for the reconstructed pixel values to differ greatly from their original values at some points. Such large-magnitude errors in the color index values result in severe color shifts at those points, which a viewer finds especially objectionable. This effect was observed even when the colormap was sorted using the technique described in Chapter 4. Therefore, transform-based PIT techniques were not considered further.

Several of the non-transform techniques found in the literature (see Chapter 2) were then investigated. A popular feature of these schemes is the use of a pyramidal (quadtree) data structure to represent the image, with decreasing spatial resolution as one nears the top of the pyramid. The techniques differ in the way the pyramid is transmitted, and also in the way

representative values are chosen for blocks of pixels on lower levels. In the terminology of the quadtree structure, the cells (pixels or representative values) in the pyramid levels are known as the *nodes* of the pyramid. The pyramid structure lends itself well to progressive transmission, and so was chosen as the basic structure for the CMPT scheme. More precisely, the CMPT pyramid has the following form:

- The bottom level of the pyramid (level 0) is the original 512×512 image.
- The top level of the pyramid (level $\log_2 512=9$) is 1×1.
- Intermediate pyramid levels (1-8) contain one-quarter the number of pixels of the previous level—image resolution is halved in both directions. The method of computing a representative value for a 2×2 block of nodes from the next lower pyramid level is to be determined.
- The transmission order for nodes in the pyramid is to be determined.

Alternatives for choosing representatives and transmitting the pyramid were then investigated for use in the CMPT scheme. These issues are considered in Section 5.3.

5.2 Objective Evaluation Criteria

Subjective evaluation methods are useful for assessing the performance of an image coding system. Such techniques were employed to choose the basic CMPT system structure. However, subjective evaluations are highly user-dependent, and do not provide a complete view of a system's strengths and weaknesses. To aid in evaluating different design alternatives for the CMPT system, it is useful to have objective measures for comparisons.

One important measure of system performance is its average coding rate, expressed as bits per pixel. To simplify efforts during the early design stages, the entropy (in bits/pixel) of a data stream was used instead of actual measured bit rates. In these cases, the assumption is that the stream will eventually be coded with some form of variable-length entropy code such as Huffman coding [38] or arithmetic coding [39]. These codes give average bit rates which approach the entropy of the source, within known performance bounds. Using the entropy measure yields performance results which are slightly better than what can be achieved with an actual system, but

eliminates the need to design a codebook for every case considered. For final evaluation of the CMPT system Huffman codes were constructed, allowing true data rates to be measured.

A measure of image quality is also needed. In order to be useful, this quality measure should agree with subjective results obtained from a human observer. As discussed in Section 3.3, perceptual color spaces are useful for defining such a measure. Two measures were chosen for use in this work. The first is the measure suggested by Frei and Baxter [25], defined using the color space developed in their work (see Section 3.3.4). The discrete form of this measure is:

$$D_{\text{Frei}}(I,\hat{I}) = \left[\frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{i=1}^{3}\left(G_i(p,q) - \hat{G}_i(p,q)\right)^2\right]^{1/2} \tag{5.1}$$

where the original image I and the reconstructed image $\hat{I}$ are $P \times Q$ images. The second quality measure is defined similarly, but is based upon the 1976 CIE L*u*v perceptual color space of Section 3.3.3:

$$D_{\text{Luv}}(I,\hat{I}) = \frac{1}{PQ}\sum_{p=1}^{P}\sum_{q=1}^{Q}\|\Gamma(p,q) - \hat{\Gamma}(p,q)\|^2$$

$$\Gamma(p,q) = \begin{bmatrix} L^*(p,q) \\ u^*(p,q) \\ v^*(p,q) \end{bmatrix}; \quad \hat{\Gamma}(p,q) = \begin{bmatrix} \hat{L}^*(p,q) \\ \hat{u}^*(p,q) \\ \hat{v}^*(p,q) \end{bmatrix} \tag{5.2}$$

Limited experiments using these two distortion measures indicate that they are in good agreement with subjective observations using the four test images.

5.3 Completing The CMPT System

In this section, we investigate ways to build upon the pyramidal structure described in Section 5.1, to form a complete progressive transmission scheme for color-mapped images. To complete the system, the method used to select representative values for the nodes of the upper pyramid levels must be defined. Also, the method used to transmit the nodes in the pyramid must

be determined. Finally, techniques to further losslessly compress the pyramid data are examined to maximize system efficiency.

5.3.1 Choosing Representative Values

Several methods of selecting representative values were considered, using the four test images. Recall that the representative value in a pyramid node represents a 2×2 block of nodes on the next lower pyramid level. The number of actual image pixels that the node represents depends on its level in the pyramid. For example, a node on pyramid level 1 represents a 2×2 pixel block, while a node on pyramid level 8 represents a 256×256 pixel block in the original image. At the receiver, an interpolation function will be used to reconstruct the pixel values from their representative value. Many of the schemes described in Chapter 2 use this technique. For example, Adelson and Burt [14] used a 5×5 gaussian convolution mask to compute the representative value. At the receiver, a similar gaussian interpolation function is used to reconstruct the pixels in the block from their representative value. Other schemes [2][19] choose the mean value of the block as a representative. At the receiver, the representative value is usually just copied into the image pixel block. This corresponds to a two-dimensional, zero-order hold (ZOH) interpolation function. Some of the authors also used higher-order interpolation functions like a first-order hold (FOH), also called bilinear interpolation. These techniques do not provide satisfactory results for CMPT, due to the index+colormap structure of the images being coded. In an achromatic image, choosing a weighted-sum representative (e.g., the mean) results in an interpolation (blending) of the pixel intensities. The viewer sees an image with less-pronounced edges, an effect which is not too objectionable in most cases. On the other hand, applying the same technique to the pixel (index) values in a color-mapped image does *not* result in an interpolation of their associated image colors. Instead, the resulting representative value is likely to point to a color that does not closely correspond to any of the pixels it is supposed to

represent. Even if the colormap has been sorted using the method of Chapter 4, this effect can still occur at the edges of the image. Tests on color-mapped images using such schemes confirm this result—the image approximations exhibit large color errors up until the lossless original is almost completely reconstructed, which requires a large amount of encoder data. These techniques do not provide the intermediate approximations needed for progressive transmission.

A pyramid scheme which does not use weighted-sum representative values is that of Dreizen [16]. In this scheme, the representative is chosen as the value of the upper-left node in the 2×2 block. At the receiver, a ZOH interpolator is used. This provides a solution for the CMPT system. Color errors will still exist in the image; however, the choice of representative ensures that at least one pixel in the block will be correctly reconstructed. Practically, more than one pixel will be well represented by this technique, since images of natural scenes tend to have highly-correlated pixels.

Before leaving the subject of representative selection, consider again the weighted-sum selection technique. In a color-mapped image, it is not generally possible to interpolate color indices to obtain interpolated colors. However, color interpolation is possible using the colormap entries themselves. This technique was not pursued for several reasons. For best results, the interpolation of two colors should be perceived as "correct" by the viewer. This implies that the interpolation should be done in a perceptual color space, such as the L*u*v* space described in Chapter 3. More importantly, the interpolation process is likely to result in a color which is not present in the colormap. Transmitting these colors would likely result in a bandwidth expansion (i.e., more data would need to be transmitted than if the original image was transmitted in a non-progressive manner), unless the color was quantized to its nearest representative in the colormap. This would require that some form of vector quantization (or at least a fast colormap search technique) be used at the transmitter. Using quantization will reduce (or eliminate) any gain in

image quality that might be achieved. The quantization operation poses an additional problem: the representative index value is no longer related to the index values of the block it represents. Most of the pyramid techniques rely on knowledge of a representative value to aid in coding later pixels efficiently. This may prevent lossless reproduction of the image unless extra data is sent, again reducing the efficiency of the system. Simply choosing the upper-left pixel as the representative value avoids these difficulties.

5.3.2 Transmitting The Pyramid Levels

Now consider how the pyramid levels are to be transmitted. One choice would be to transmit each level in succession, starting with the top of the pyramid. At the receiver, each level would be expanded to the original image size using, for example, a ZOH interpolator. This homogeneous transmission method does not take into consideration the content of the image. One of the goals of PIT is to provide early recognition of an image. Therefore, it would be more appropriate to develop the pyramid in a nonhomogeneous manner which decides when to transmit a pyramid node based on an estimate of the information it contains. For the CMPT system, the technique for transmitting the pyramid is:

1. For each progressive pass desired, choose a pass threshold parameter P.
2. Starting with the node at the top of the pyramid, compute an information measure, H, for the block of pixels which that node represents. If $H < P$, skip to Step 4.
3. If $H \geq P$, expand the node at the receiver into the 2×2 block of nodes it represents, and reconstruct the pixels accordingly.
4. Proceed to the next eligible node. The next eligible node is a node on the same pyramid level which has not yet been expanded. After checking all eligible nodes on the current pyramid level, continue by checking nodes on the next lower level until all nodes are checked.
5. After all eligible nodes in the pyramid are tested, the progressive pass is complete and the next pass begins at Step 2 with a new value for the threshold P, until the desired number of progressive passes has been transmitted or image has been transmitted losslessly.

In this way, each progressive pass will be composed of nodes from several different pyramid

levels. The effect at the receiver is that the reconstructed image will consist of several different sizes of pixel blocks, with smaller blocks used in areas which are considered to have more information by the algorithm. Several methods of computing the information measure were examined, and the results are discussed below.

In his work toward progressive transmission of achromatic images, Dreizen [16] examined several methods for estimating the information in a pixel block. His choice was a contrast-like measure, which is simple to compute and performs adequately for controlling nonhomogeneous progressive transmission. The information measure, H, is defined as:

$$H = I_{max} - I_{min} \tag{5.3}$$

where I_{max} and I_{min} are the largest and smallest pixel values in the block, respectively. With this definition and an 8-bit image, H will always be an integer in the interval [0, 255]. Dreizen chose the set of pass parameters (thresholds) $\{P_k\} = \{64, 32, 16, 8, 4, 2, 1\}$. This choice yields seven progressive passes, and the choice of P=1 for the final pass ensures that the final image will be a lossless reproduction of the original image.

The above information measure was then modified for use in the CMPT system. One variation of the measure uses the RGB values of the colormap directly. It is defined as:

$$H_{RGB} = \max(H_R, H_G, H_B) \tag{5.4}$$

where H_R, H_G, and H_B are defined in a similar manner to Dreizen's original H, that is, by subtracting the maximum and minimum red, green, or blue components of the pixels in the block, respectively. Since the R, G, and B entries of the colormap are 8-bit integers with values in the range [0, 255], the pass parameters P_k are also integers: {128, 80, 60, 40, 20, 15, 10, 1}. As with the original measure, the final pass parameter is 1, which gives lossless transmission of the original image.

Another information measure used for the CMPT system was defined using the CIE L*u*v* space:

$$H_{Luv} = \max(H_L, H_u, H_v) \tag{5.5}$$

where H_L, H_u, and H_v are defined in a similar manner as H_{RGB}, using the maximum and minimum color components in L*u*v* space. A third measure used is a variation on H_{Luv}:

$$H_{Luv2} = \sqrt{H_L^2 + H_u^2 + H_v^2} \tag{5.6}$$

The pass parameters for the last two measures must be chosen differently than the first, since the values of $H_{(\cdot)}$ are no longer integers. The parameter set chosen was {30.0, 20,0, 10.0, 5.0, 2.5, 1.25, 0.0}. The last pass parameter is chosen as zero to ensure lossless final transmission.

Other choices of the information measure are possible. For example, one could use the entropy of the pixels in the block as the information measure. Dreizen also suggests an RMS minimization technique developed by Sans, et al.[40], or Chang and Yang's PIM measure [41] as alternatives. Rost [42] has also developed a variable-blocksize PIT scheme which uses a distortion measure between the original and reconstructed pixel block to determine when a block should be transmitted. The simple max-min measure chosen for the CMPT system has the advantage that it is simple to compute, and provides reasonable performance. Of the different measures tested, the RGB measure is simplest, since no nonlinear transformations of the colormap entries is required. Intuitively, using a perceptual color space such as the L*u*v* space should provide better perceptual quality than an RGB-based information measure for a given number of transmitted blocks; however, preliminary tests of this hypothesis were inconclusive. Nevertheless the measure H_{Luv} was chosen for use in the final CMPT system.

The choice of pass parameters given above is not critical. They must, however, satisfy a few simple constraints. Since the information measures used are nonincreasing as the pyramid

nodes are transmitted, the pass parameters should also decrease for later passes. If a lossless final image is desired, the final pass parameter should be zero or one, depending on which of the above measures is used. The number of pass parameters chosen will determine the number of progressive approximations transmitted. The values of the pass parameters can be adjusted to provide (approximately) any coding rate or image quality within reasonable limits.

Allowing nonhomogeneous pyramid transmission requires additional transmission overhead so that the transmitter can inform the receiver whether or not a pyramid node is to be transmitted. Both the transmitter and receiver have knowledge of the pass parameters, and traverse the pyramid nodes in the same manner. Therefore, a single "1" bit can be used to signal the receiver that a node is about to be transmitted, and "0" bit will indicate that the node is to be skipped for that pass. These bits are called node selection bits (NSBs). If a one is transmitted, it will be followed by coded values which will be used to expand the pyramid node into the 2×2 block of nodes it represents. The encoding of this data is discussed in Section 5.3.3. Several thousand NSBs may be transmitted for a 512×512 image: one for each expanded pyramid node, and many others for nodes which are skipped during progressive passes. This overhead can be considerable, about 0.58 bits/pixel for the test images; however, the NSB data stream can be further compressed through the use of entropy coding. This is the subject of Section 5.3.4.

An additional modification to the pyramid structure was made to improve the subjective quality of early transmission passes. In its original form, the pyramid-based scheme transmits the top level of the pyramid first, a single value which represents the entire image (512×512 pixels for this work). Nodes on the next pyramid level represent 256×256 pixel blocks, etc. At the receiver these pyramid levels produce an extremely coarse image approximation, which is of little or no use to the viewer. For low-information blocks, several progressive passes may occur before a large block (e.g., 64×64) is subdivided. This blocking effect is very noticeable. Instead, the

pyramid was modified to present its first approximation using 32×32 blocks. This corresponds to discarding levels 6-9 of the pyramid. The result is a more useful initial approximation, with a slight savings in transmission bandwidth since no NSBs need to be transmitted. Subsequent image approximations are constructed using the nonhomogeneous algorithm described previously. The 32×32 block size was considered appropriate for the 512×512 images used for this work. For images of a different size, a different block size might be preferable.

5.3.3 Encoding The Pyramid Nodes

When the information measure for a block is greater than or equal to the pass parameter, a node selection bit of one is transmitted, followed by additional data. This data will be used to expand the pyramid node into the 2×2 block of nodes it represents. In the reconstructed image, this corresponds to replacing an $M \times M$ block of pixels with four pixel blocks of size $\frac{1}{2}M \times \frac{1}{2}M$. The four pixel blocks are reconstructed at the receiver using a ZOH interpolator. Therefore, four color index values are required to expand the block. This situation is illustrated in Figure 5.1.

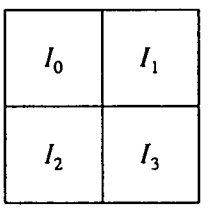


Figure 5.1: Block of Pyramid Nodes (2×2)

The index value for the upper-left block, I_0, is already known, since by virtue of the encoding algorithm it is the representative value of the entire block before expansion. The three remaining values will be sent using a lossless predictive coder. Such a coder will provide efficient data compression if there some redundancy (correlation) among the image pixels [1]. Recall from Chapter 4 that although this is not normally the case for color-mapped images, pixel correlation can be restored by sorting the image colormap. For greatest compression, the images presented

to the CMPT coder should have this sorting performed on them.

The predictive coder functions by transmitting the error between the value to be transmitted, I_k, and a predicted value, J_k. The predictors chosen are identical to those of Dreizen, which require only the four values in the block to be expanded at the transmitter and are given by:

$$\begin{aligned} J_1 &= I_0 \\ J_2 &= \left\lfloor \frac{1}{2}(I_0 + I_1) \right\rfloor \\ J_3 &= \left\lfloor \frac{1}{2}(I_1 + I_2) \right\rfloor \end{aligned} \tag{5.7}$$

More complex predictor functions could be used, and there has been considerable research toward finding optimum, or good sub-optimum, predictors [1]. However, putting a large effort toward designing a better predictor using the I_k values may not greatly increase performance, since the values used in the predictor may have a large spatial separation. A distant pixel will not be well-correlated with the pixel to be transmitted. For example, when expanding a 32×32 block into four 16×16 blocks, the values of I_1 and I_2 are spatially separated by a distance of (1.414)(16) pixels. More effort could be made toward locating spatially closer pixels for use in the predictor, but a closer pixel may not be available. The simple predictors used here perform adequately for the CMPT system.

The prediction errors using this scheme will be integers. If the prediction is good, the prediction errors will be approximately zero mean, with a distribution which is clustered around zero. Further compression can then be achieved by using some form of entropy coding. For the CMPT scheme, Huffman coding [38] will be used. First, it will be useful to "precondition" the prediction errors to make them easier to encode. This conditioning is based on the following observation. For an image with 8-bit color index values, the prediction errors will be integers in

the range [-255, 255], an alphabet size of 511. However, for a given predictor value, J, there are only 256 possible differences (one for each of the 256 possible I values). Since the receiver always knows the predictor value during transmission, only 256 codewords are needed to represent the difference value. Designing a Huffman code for each of the 256 different predictor values is costly, since a large amount of storage would be required for the 256 codebooks of 256 entries each. Instead, a single Huffman codebook with 256 entries is created. To be as efficient as possible, the codeword for a difference value of zero should be the same for every possible predictor value, since it will (hopefully) be the most frequently used symbol regardless of the predictor's value.

Dreizen gives one possible implementation of this idea. The technique is to "fold" the difference values around zero. For example, if the predictor value is J=2, the set of possible differences

0, +1, -1, +2, -2, +3, +4, +5, +6, ..., +253

are mapped in a one-to-one manner onto the set of values

0, +1, -1, +2, -2, +3, -3, +4, -4, ..., +127, -127, +128

The Huffman code is then designed based on the distribution of the mapped difference values. In the CMPT scheme, this difference-mapping function will be used if the image colormap has been sorted as a linear list structure (see Section 4.5).

If the colormap was considered as a circular ring structure, a different mapping function will be used. This is necessary since the colors with indices of 0 and 255 (assuming an 8-bit colormap) are now considered "close". That is, if the value to be transmitted is 255 and the predicted value is 0, the magnitude of the prediction error should be 1, not 255. The remapping algorithm is simple:

```
E = I - J                (the prediction error)
If (E < -127) Then
   E = E + 256
Else If (E > 128) Then
   E = E - 256
Else
   (no change)
```

The Huffman code is then designed based on the remapped difference values, which are always in the interval [-127, +128]. Either remapping technique may be used in a CMPT system, depending on which colormap sorting method has been used. Using a Huffman code with a difference mapping technique, the simulated CMPT scheme was able to code the node representative values at an average rate of 6.23 bits/value.

5.3.4 Encoding The Node Selection Bits

The node selection bits can add a significant amount of overhead to the transmission process, approximately 0.58 bits/pixels for the test images used in this work. The number of NSBs required will vary with image content and the choice of pass parameters. The NSBs cannot easily be run-length encoded [1] since only very short runs exist in the bit stream. Instead, a predictive scheme can be used. The predicted value for the current NSB is chosen as the previous NSB. If they are the same, a zero bit is transmitted. If they are different, transmit a one bit. The technique is simply modulo-2 addition, and the resultant stream will contain a large number of zero bits.

An entropy code can then be developed to efficiently code these bits. A Huffman entropy code was used for this task. However, a Huffman code will not be effective if the alphabet to be coded is {0, 1}. This is due to the fact that the Huffman algorithm can only assign an integer number of bits to each symbol in the input alphabet. In this case, a single bit would need to be assigned to each letter of the alphabet, with a result of no coding gain. To overcome this

limitation, multiple bits can be grouped together in fixed-length quantities, and a code can then be designed for this new alphabet. For this work, the output bits of the predictive NSB coder were grouped into 8-bit quantities, and a 256-entry Huffman codebook was designed. With this code, the NSBs can be coded at an average rate of 6.72 bits/8-bit group, or 0.84 bits/NSB.

5.4 Simulation Results

The CMPT system developed in this thesis was simulated using the four test images as inputs. In this section, the results of these simulations will be evaluated.

All simulations were written in the C programming language and were executed on a Sun Microsystems SPARCstation 1, a UNIX-based workstation. The simulator provides a real-time view of the progressive transmission process, which was implemented using the X Window System from MIT. This indicates that the complexity of the algorithm is within the capabilities of currently available equipment. Unlike many pyramid-based PIT schemes, the CMPT coder has the added advantage that the pyramid structure does not need to be explicitly built in memory. This is a direct result of choosing the upper-left pixel as representative. The major storage requirements of the CMPT encoder (or decoder) are:

- An image buffer, 512^2 locations for this work
- Storage for the colormap, 3×256 locations for this work
- The Huffman codes for transmitting the NSBs and representatives, two codebooks of 256 codewords each
- An array of bits to indicate which pyramid nodes have been transmitted, approximately 11K bytes for this work

Since the pyramid does not need to be built, there is also no encoding delay associated with the CMPT scheme as might be encountered with other pyramid schemes.

Two simulation runs for each image are presented. One set of simulations uses the RGB-based information measure, H_{RGB}, described in Section 5.3.2. The second set of simulations uses the L*u*v*-based information measure, H_{Luv}. In each simulation, the pass parameters are chosen

so that the final progressive pass is a lossless reproduction of the original image.

Simulations were also conducted to compare the performances of the two difference-mapping methods described in Section 5.3.3. Results of these simulations show that both difference-mapping techniques provide similar data rates. Therefore, only a single set of simulation results will be presented, those using the folded-difference mapper. For use with this difference mapper, all images have had their colormaps sorted as a linear list. In all cases, the colormap sorting was performed in the CIE L*u*v* color space. The simulation results are given in Sections 5.4.1 and 5.4.2.

A possible drawback of progressive transmission is its potential for bandwidth expansion when an image is losslessly transmitted. Therefore, the CMPT coder was also compared with a standard full-frame image compression algorithm for color-mapped images. The results are given in Section 5.4.3.

5.4.1 Results Using An RGB-Based Information Measure

Table 5.1 shows the bit rates and distortion measures for each pass of the CMPT coder when the information measure is based on an RGB-color space. The distortion measures are those previously defined in Section 5.2. The bit rates are the actual coding rates obtained from the simulator. This rate includes all data required to transmit the image except for what is required to transmit the colormap. The amount of data required to transmit the colormap is at most 3×256×8=6144 bits, an overhead of 0.0234 bits/pixel for a 512×512 image. This overhead is constant for all the images. The values of P_k are the pass parameters (information measure thresholds) used for transmitting the image.

With the chosen pass parameters, the CMPT scheme provides images with a steadily improving quality, while still maintaining a low data rate for the early passes. Image content is visible almost immediately, especially in areas which contain a significant number of edges.

Table 5.1: CMPT Using An RGB-Based Information Measure

		Hat			Park		
Pass	P_k	Rate	D_{Frei}	D_{Luv}	Rate	D_{Frei}	D_{Luv}
0	128	0.14	2.50	33.82	0.47	2.43	26.99
1	80	0.44	1.58	11.85	1.14	1.69	12.86
2	60	0.71	1.26	7.80	1.70	1.35	7.97
3	40	1.27	0.90	3.71	2.67	0.99	4.26
4	20	3.06	0.51	1.04	5.15	0.42	0.64
5	15	4.53	0.32	0.38	6.12	0.24	0.18
6	10	5.85	0.13	0.05	6.79	0.09	0.02
7	1	6.32	0	0	7.04	0	0

		Omaha			Lincoln		
Pass	P_k	Rate	D_{Frei}	D_{Luv}	Rate	D_{Frei}	D_{Luv}
0	128	0.15	2.60	22.26	0.02	1.79	14.66
1	80	0.60	1.90	11.87	0.09	1.61	11.25
2	60	1.14	1.50	7.64	0.22	1.45	8.81
3	40	2.41	0.99	3.43	0.71	1.08	5.19
4	20	4.50	0.49	0.83	2.24	0.68	2.11
5	15	5.73	0.27	0.23	4.33	0.40	0.68
6	10	5.99	0.23	0.16	4.39	0.39	0.67
7	1	6.86	0	0	6.68	0	0

These edges allow the eye to distinguish the various parts of each image more quickly. Very good to excellent quality images are obtained by Pass 3 of the algorithm. Images with more detail, such as the Park image, have a slightly faster increase in data rate since more blocks are being transmitted. In each case the image of Pass 7 is a lossless reproduction, and compression of the original 8-bit image is still achieved.

Note that the coding rates for the Lincoln image are particularly low for the first several

passes, and that the distortions of the reconstructed images do not decrease as rapidly with each progressive pass as those of the other images. The Lincoln image is a particularly low-contrast image. However, it contains an aerial view of many city blocks, which have a number of high-detail pixel blocks (see also Figure 4.8). This makes it particularly troublesome for the CMPT coder. Few blocks are transmitted during early passes, followed by a large burst of blocks when the pass parameter becomes sufficiently small. Some suggestions for preventing this effect are discussed in Section 5.5.

5.4.2 Results Using An L*u*v*-Based Information Measure

Table 5.2 shows the bit rates and distortion measurements for each pass of the CMPT coder when the information measure is based on the CIE L*u*v* perceptual color space. The distortion measures are those defined in Section 5.2. The data required to transmit the image colormap is 0.0234 bits/pixel for a 512×512 image. Figures 5.2 through 5.7 show an example of passes 0-5 of the CMPT process, using the Hat image. Pass 6 is omitted, since it is a lossless reproduction of the original image (see Figure 4.2).

The results of using an L*u*v*-based information measure are similar to those obtained for the RGB-based measure. The CMPT scheme again provides detail early in the transmission process, especially in edge areas of the image. Excellent quality reconstructed images are obtained after only a few passes, at rates of 2-3 bits/pixel. As in the case of the RGB-based measure, the Lincoln image exhibits particularly low initial rates. However, the pass parameters chosen for the L*u*v* scheme seemed to provide a more gradual increase in data rate.

Perceptual quality of the images from the two simulations is also very similar. Efforts to determine which information measure provides better subjective image quality were made, but no definite results were obtained.

Table 5.2: CMPT Using An L*u*v*-Based Information Measure

		Hat			Park		
Pass	P_k	Rate	D_{Frei}	D_{Luv}	Rate	D_{Frei}	D_{Luv}
0	30.00	0.16	2.48	23.34	0.39	2.72	31.30
1	20.00	0.40	1.78	12.47	0.86	1.94	15.08
2	10.00	1.06	1.08	4.29	2.31	1.16	4.69
3	5.00	2.21	0.65	1.28	4.22	0.62	0.95
4	2.50	3.42	0.45	0.47	5.24	0.38	0.28
5	1.25	4.54	0.30	0.17	5.89	0.27	0.12
6	0	6.29	0	0	7.09	0	0

		Omaha			Lincoln		
Pass	P_k	Rate	D_{Frei}	D_{Luv}	Rate	D_{Frei}	D_{Luv}
0	30.00	0.13	2.71	26.21	0.02	1.79	14.00
1	20.00	0.40	2.14	15.63	0.11	1.60	10.88
2	10.00	1.78	1.30	5.01	0.71	1.13	5.23
3	5.00	3.98	0.65	1.14	2.59	0.67	1.67
4	2.50	5.35	0.37	0.33	4.56	0.40	0.52
5	1.25	6.15	0.21	0.10	5.97	0.19	0.10
6	0	7.07	0	0	6.87	0	0

5.4.3 Comparison of Lossless Transmission Rates

Progressive transmission schemes have a potential disadvantage from a compression standpoint. By requiring that the coder be able to present successive approximations of an image, some of its effectiveness as a data compressor may be lost. In addition, the nonhomogeneous approach of the CMPT scheme requires additional data (the NSBs) to encode the image. A full-frame compression algorithm does not have these limitations. Therefore, it is useful to examine the cost of progressively transmitting the image.

In Table 5.3, the CMPT results for lossless image transmission are compared to those of a widely-available, full-frame coder for color-mapped images. The full-frame scheme is the GIF image interchange format [33]. This image format consists of header information plus a compressed image data stream. For compression, the GIF format uses a modified Ziv-Lempel technique known as the LZW [43] algorithm. An implementation of this algorithm is also available from the popular UNIX *compress* program. To facilitate comparison, the GIF header information must be separated from the compressed data. Since the compressed image data is essentially identical to the output of *compress*, the output of that program will be used as the basis for comparison. The rates in Table 5.3 are for the 512×512 image array only; no colormap data was included.

Table 5.3: Lossless Compression Comparison

Image	CMPT Rate	GIF Rate
Hat	6.29	5.94
Park	7.09	6.48
Omaha	7.07	6.51
Lincoln	6.87	5.30

For three of the images tested the CMPT coding rates approach those of the lossless full-frame coder, adding an average of 0.5 bits/pixel over the full-frame coder. This makes the CMPT coder a possible alternative to a full-frame coder, especially when its progressive nature is also considered. The higher CMPT rate observed for the Lincoln image can be again be attributed to its low contrast and high-frequency content. Considering all four images together, the rate of the CMPT coder is an average of 0.77 bits/pixel greater than that of the full-frame coder.

5.5 Possible Modifications and Improvements

During the design of the CMPT system, many alternatives were available for its components. Some of these options were investigated, but others were omitted due to time and resource constraints. In this section some of these possibilities will be discussed, and may be used as a starting point for future research. Many of these ideas can be applied to all types of progressive transmission, not just the CMPT system developed in this thesis.

Modifications could be pursued to improve the compression efficiency of the CMPT coder. In Section 5.3, it was noted that the predictive scheme used to encode the pyramid nodes was rather simplistic. It is likely that a more in-depth analysis of the pyramid data could yield a more efficient predictive scheme. For the node selection bits, an improvement in compression should also be realizable. A good approach to this problem would be to formulate a more detailed model for the NSBs. The NSBs exhibit somewhat stationary statistics over short durations, for which a Markov process model might be appropriate [1].

Many changes could be made to the CMPT coder to improve its ability to present good quality images to the viewer. For example, a different method of selecting the pyramid nodes for transmission might be used. In Section 5.3, several alternatives for the information measure, H, were mentioned. A good measure would be one which takes into account the way a viewer perceives the reconstructed image. Such a measure might take into account the spatial dependencies of the human visual system. Also, the pass parameters could be adaptively selected based on the content of the image. These techniques could also help to avoid poor coding performance, as in the case of the Lincoln test image.

If a progressive transmission coder is used for a particular type of image, further improvements may be possible. For example, consider a database of facial images. The goal of

a PT system in this application would be to provide early recognition of facial features. A reasonable modification to the nonhomogeneous pyramid transmission would be the following. Introduce an additional weighting factor which is dependent on the spatial position of the block under consideration. In this way, the coder will spend more effort on areas of interest to the user, i.e., the central portion of the image.

In Chapter 4, it was observed that sorting an image colormap offered two main advantages. The first was that the color index values became more correlated, a feature that was used to improve the coding efficiency of the CMPT system. The second advantage of colormap sorting is that small errors in the image become tolerable for the viewer. That is, some degree of quantization can now be performed on the image to achieve further compression. However, the amount of tolerable error is still small. For a typical quantizer, this presents a problem. Inputs which fall into the inner quantizer levels will have bounds on the magnitude of the quantization error [1]. However, inputs that are outside the minimum and maximum quantizer decision levels have no upper bound on the error magnitude, a condition known as slope-overload noise. Recently, a quantization technique was developed by Sayood and Na [44], which eliminates slope-overload noise. This technique, called recursively-indexed quantization, removes the slope-overload quantization noise at the expense of an increase in encoder data. Performance of the technique is comparable to more complex techniques such as vector quantization, with the added advantage that a quantizer can now be designed with guaranteed limits on the quantization error. Such a quantizer could be used in a CMPT coder to allow limited errors to be introduced in the image during transmission. The final transmitted image would no longer be a lossless copy of the original, but transmission time would be reduced. In a CMPT system this technique might be made available as an option, to allow the user to trade image quality in favor of transmission speed according to his or her needs.

A final improvement to the CMPT system is aimed at improving image quality at the receiver. Early approximations from many progressive transmission schemes have a pronounced blocking effect. In the CMPT scheme, the zero-order hold interpolator used is responsible for this effect. To improve image quality, higher order interpolation functions might be used. For example, bilinear (first-order hold) interpolation could be used, or perhaps a more complex function such as a spline surface [1][45]. In addition, examples of more complex post-processing techniques may be found in the literature [40][46]. The effectiveness of these techniques is dependent on the user's display hardware. For example, if the transmitted image uses an 8-bit colormap but the user's display system has a 24-bit display system (see Section 2.4), an interpolation or smoothing algorithm could enhance the image quality by adding additional colors to the image which are not found in the original colormap. If the user's display is an 8-bit, color-mapped frame buffer, however, no additional colors for the image are available. In this case, another possible option is to use dithering [1][27], the addition of controlled amounts of noise to the image to enhance its subjective quality. In more advanced display systems, some or all of these features may be performed by dedicated hardware. In any case, post-processing can still be implemented as an option for the user, who can then choose to use it based on his or her needs and the capabilities of the display equipment. Since all of this processing is done at the receiver, no additional data needs to be transmitted to support it. This can be important in image database applications since no additional load is placed on the central host, which may be required to serve many users.

In this chapter, the results of simulating the CMPT scheme of this thesis were presented. The CMPT scheme provides a workable method for progressive transmission of color-mapped image data. In addition, the rates obtained for lossless transmission of the test images makes it

competitive with a full-frame color-mapped image compression technique. Finally, several modifications to the current scheme were presented which could be used to improve the current scheme.

Figure 5.2: CMPT Hat Pass 0, 0.16 bpp

Figure 5.3: CMPT Hat Pass 1, 0.40 bpp

Figure 5.4: CMPT Hat Pass 2, 1.06 bpp

Figure 5.5: CMPT Hat Pass 3, 2.21 bpp

Figure 5.6: CMPT Hat Pass 4, 3.42 bpp

Figure 5.7: CMPT Hat Pass 5, 4.54 bpp

6. Summary and Conclusions

Chapters 1 through 4 of this thesis are intended to provide background information for the coder developed in Chapter 5. Material in these chapters includes progression transmission techniques, and the part of visual perception humans know as "color". Also included is information about modern display hardware, to provide an introduction to pseudo-color or color-mapped images and motivate the CMPT coder developed in this work. Several methods for creating a color-mapped image from a full-color original are also presented.

In order to construct an efficient CMPT scheme, analysis was performed on color-mapped image data. Some interesting properties of color-mapped images were noted, and methods of processing the data are presented to allow the images to be coded more efficiently. In particular, the colormap sorting technique presented in Section 4.5 restores the correlation between the image's color index values. This allows standard predictive coding techniques to be used effectively to compress the image. The sorting operation also allows controlled amounts of error to be introduced in the image without drastic changes in image quality, such as might be introduced by quantization.

In Chapter 5, the ideas of previous chapters are combined to construct the CMPT coder. The coder uses a modified pyramid structure as the basis for progressive transmission, and transmits the nodes of the pyramid in a nonhomogeneous manner based on an information measure. In this manner, areas with a high level of detail are refined more quickly than areas of lower detail. As implemented, the CMPT coder also provides a lossless reproduction of the original image as its final pass, but could provide a lossy final image if necessary. Through the use of a lossless predictive coder plus entropy coding, the CMPT coder provides lossless

compression rates which approach those of a non-progressive, full-frame coder for color-mapped images. The overall complexity of the scheme is low, and requires little extra memory to manage the pyramid data beyond what is needed to hold the original and reconstructed image pixels themselves. The CMPT simulator can run in real-time on a workstation platform, indicating that the scheme is within the capabilities of currently available computing equipment. Suggestions for improving the quality of the early transmission passes are presented, and may be implemented as a user-selectable option to the coder. Quality improvement is obtained without the need for additional channel resources.

In conclusion, the CMPT coder developed in this work provides a feasible solution for progressively transmitting color-mapped images. With this technique, users will benefit from both the effective compression available with PIT, as well as actual compression when the image is losslessly reconstructed. During transmission, the coder is able to locate areas of high detail and present them to the user more quickly. This allows the user to assess the image more quickly. In addition, an analysis of color-mapped images was performed. The results of this analysis, and the solutions that resulted from it, should also be useful for designing other coders for this type of image data, progressive or not.

References

[1] Jayant, N.S., and P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ: Prentice-Hall, 1984.

[2] Sloan, K.R., and S.L. Tanimoto, "Progressive Refinement of Raster Images," *IEEE Transactions on Computers*, vol. C-28, no. 11, pp. 871-4, November 1979.

[3] Knowlton, K., "Progressive Transmission of Grey-Scale and Binary Pictures by Simple, Efficient, and Lossless Encoding Schemes," *Proceedings of the IEEE*, vol. 68, no. 7, pp. 585-896, July 1980.

[4] Rao, K.R., and P. Yip, *Discrete Cosine Transform*, Boston, MA: Academic Press, Inc., 1990.

[5] Tzou, Kou-Hu, "Progressive Image Transmission: A Review and Comparison of Techniques," *Optical Engineering*, vol. 26, no. 7, pp. 581-589, July 1987.

[6] Goldberg M., and L. Wang, "Comparative Performance of Pyramid Data Structures for Progressive Image Transmission," *IEEE Trans. on Communications*, vol. 39, no. 4, pp. 540-547, April 1991.

[7] Ngan, K., "Image Display Techniques Using the Cosine Transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 173-177, February 1984.

[8] Dubois, E., and J.L. Monchet, "Encoding and Progressive Transmission of Still Pictures in NTSC Composite Format Using Transform Domain Methods," *IEEE Trans. Comm.*, vol. COM-34, no. 3, pp. 310-319, March 1986.

[9] Chen, W.H., and W.K. Pratt, "Scene Adaptive Coder," *IEEE Trans. Comm.*, vol. COM-32, no. 3, pp. 225-232, March 1984.

[10] Chitprasert, B., and K.R. Rao, "Human Visual Weighted Progressive Image Transmission," *IEEE Trans. Comm.*, vol. 38, no. 7, pp 1040-44, July 1990.

[11] Chen, W.H., and H. Smith, "Adaptive Coding of Monochrome and Color Images," *IEEE Trans. Comm.*, vol. COM-25, no. 11, pp. 1285-92, Nov. 1977.

[12] Frank, A.J., J.D. Daniels, and D.R. Unangst, "Progressive Image Transmission Using A Growth-Geometry Coding," *Proceedings of the IEEE*, vol. 68, no. 7, pp. 897-909, July 1980.

[13] Hill, F.S., S.W. Walker, and F. Gao, "Interactive Image Query System Using Progressive Transmission," *Comp. Graphics*, vol. 17, no. 3, pp. 323-30, July 1983.

[14] Burt, P.J., and E.H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Trans. Comm.*, vol. COM-31, no. 4, pp. 532-540, April 1983.

[15] Hoffman, W.D., and D.E. Troxel, "Making Progressive Transmission Adaptive," *IEEE Trans. Comm.*, vol. COM-34, no. 8, pp. 806-813, August 1986.

[16] Dreizen, H., "Content-Driven Progressive Transmission of Grey-Scale Images," *IEEE Trans. Comm.*, vol. COM-35, no. 3, pp. 289-296, March 1987.

[17] Wang, L., and M. Goldberg, "Progressive Transmission Using Vector Quantization on Images in Pyramid Form," *IEEE Trans. Comm.*, vol. 27, no. 12, pp. 1339-1349, December 1989.

[18] Linde, Y., A. Buzo, R.M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Comm.*, vol. COM-28, no. 1, pp. 84-95, January 1980.

[19] Wang, L., and M. Goldberg, "Reduced-Difference Pyramid: A Data Structure for Progressive Image Transmission," *Optical Engineering*, vol. 28, no. 7, pp. 708-16, July 1989.

[20] Wysecki, G. and W.S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae, Second Edition*, New York: John Wiley and Sons, 1982.

[21] *Book of Color*, Munsell Color Company, Baltimore, MD., 1929.

[22] Jain, A.K., *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[23] Limb, J.O., C.B. Rubinstein, and J.E. Thompson, "Digital Coding of Color Video Signals—A Review," *IEEE Trans. Comm.*, vol. COM-25, no. 11, pp. 1349-1384, November 1977.

[24] CIE Colorimetry Committee, "Proposal for Study of Color Spaces and Color-Difference Evaluations," *Jour. Opt. Soc. of Amer.*, vol. 64, pp. 896-7, June 1974.

[25] Frei, W., and B. Baxter, "Rate-Distortion Coding Simulation for Color Images," *IEEE Trans. Comm.*, vol COM-25, no. 11, pp. 1385-1392, November 1977.

[26] Faugeras, O.D., "Digital Color Image Processing and Psychophysics Within the Framework of a Human Visual Model," (Ph.D. Dissertation), University of Utah, June 1976.

[27] Foley, J.D., van Dam, A., S.K. Feiner, and J.F. Hughes, *Computer Graphics: Principles and Practice (Second Edition)*, Reading, MA: Addison-Wesley, 1990.

[28] Barkakati, N., *X Window System Programming*, Carmel, IN: Howard Sams Co., 1991.

[29] Heckbert, P., "Color Image Quantization for Frame Buffer Display," *Computer Graphics*, vol. 16, no. 3, pp. 297-307, July 1982.

[30] Wan, S.J., S.K.M. Wong, and P. Prusinkiewicz, "An Algorithm for Multidimensional Data Clustering," *ACM Transactions on Mathematical Software*, vol. 14, no. 2, pp. 153-162, June 1988.

[31] MacQueen, J.B., "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability Volume 1*, pp. 281-297, 1967.

[32] Selim, K., and M.A. Ismail, "K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality," *IEEE Trans. Pat. Analysis and Mach. Int.*, vol. PAMI-6, no. 1, pp. 81-86, Jan. 1984.

[33] *Graphics Interchange Format (GIF) Specification*, CompuServe, Inc., Columbus, OH, June 1987.

[34] Aarts, E., and J. Korst, *Simulated Annealing and Boltzmann Machines*, New York: John Wiley and Sons, 1989.

[35] Lin, S., "Computer Solutions of the Traveling Salesman Problem," *Bell System Technical Journal*, pp. 2245-2269, December 1965.

[36] Bellman, R.E., and S.E. Dreyfus, *Applied Dynamic Programming*, Princeton, NJ: Princeton University Press, 1962.

[37] Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C*, New York: Cambridge University Press, 1988.

[38] Huffman, D.A., "A Method for the Construction of Minimum Redundancy Codes," *Proceedings of the IRE*, vol. 40, pp. 1098-1101, September 1952.

[39] Bell, T.C., J.G. Cleary, and I.H. Witten, *Text Compression*, Englewood Cliffs, NJ: Prentice-Hall, 1990.

[40] Sanz, A., C. Muñoz, and N. García, "Approximation Quality Improvement Techniques in Progressive Image Transmission," *IEEE Journal on Selected Areas in Communications*, vol. SAC-2, no. 2, pp. 359-373, March 1984.

[41] Chang, S.K., and C.C. Yang, "Picture Information Measures for Similarity Retrieval," *Comp. Vision, Graphics, and Image Proc.*, vol. 23, pp. 366-375, 1983.

[42] Rost, M.C., "Data Compression Using Adaptive Transform Coding," (Ph.D. Dissertation), University of Nebraska, October 1988.

[43] Welch, T.A. "A Technique for High-Performance Data Compression," *IEEE Computer*, pp. 8-19, June 1984.

[44] Sayood, K., and S. Na, "Recursive Quantization," *Proc. Twenty-Fourth Asilomar Conference on Signals, Systems, and Computers*, 1990.

[45] Gonzales, R.C., and P. Wintz, *Digital Image Processing (Second Edition)*, Reading, MA: Addison-Wesley, 1987.

[46] Ramamurthi, B., and A. Gersho, "Nonlinear Space-Variant Postprocessing of Block Coded Images," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 5, pp. 1258-1267, October, 1986.

[47] Papadimitriou, C.H., and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Englewood Cliffs, NJ: Prentice-Hall, 1982.