Color Quantization of Images [†‡]

Michael Orchard
Department of Electrical Engineering
Princeton University
Princeton, NJ 08544
(609) 258-4643

Charles Bouman
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907-0501
(317) 494-0340

Abstract [1]

Many image display devices allow only a limited number of colors to be simultaneously displayed. Usually, this set of available colors, called a color palette, may be selected by a user from a wide variety of available colors. Such device restrictions make it particularly difficult to display natural color images since these images usually contain a wide range of colors which must then be quantized by a palette with limited size. This color quantization problem is considered in two parts: the selection of an optimal color palette and the optimal mapping of each pixel of the image to a color from the palette.

This paper develops algorithms for the design of hierarchical tree structured color palettes incorporating performance criteria which reflect subjective evaluations of image quality. Tree structured color palettes greatly reduce the computational requirements of the palette design and pixel mapping tasks, while allowing colors to be properly allocated to densely populated areas of the color space. The algorithms produce higher quality displayed images and require less computations than previously proposed methods.

Error diffusion techniques are commonly used for displaying images which have been quantized to very few levels. This paper studies problems related to the application of error diffusion techniques to the display of color images. A modified error diffusion technique is proposed for resolving these problems. The new error diffusion techinque is shown to be easily implemented using the tree structured color palettes developed earlier.

EDICS #6.2.6 Image Transformations for Display and Printing

[†]This work was supported by the Office of Naval Research and the Naval Research Laboratory under Grant N00014-87-0189, by an IBM Graduate Fellowship, and by an AT&T Fellowship.

[‡]*IEEE Trans. on Sig. Proc.*, vol. 39, no. 12, pp. 2677-2690, Dec. 1991.

[1]Permission to publish this abstract separately is granted.

1 Introduction

Video monitors display color images by modulating the intensity of the three primary colors (red, green and blue) at each pixel of the image. In a digitized color image each primary color is usually quantized with 8 bits of resolution in order to eliminate distinguishable quantization steps in luminance, color hue and color saturation. Thus, full color digital display systems use 24 bits to specify the color of each pixel on the screen.

The cost of high-speed memory needed to support such a full-color display on a high-resolution monitor makes many applications impractical. An alternative approach, which is used by many currently available displays, is to provide a limited number of bits, such as 8, for specifying the color at each pixel. Each of these $2^8 = 256$ values is then used as an index into a user defined table of colors. Each entry into the table contains a 24 bit value which specifies the color's red, green and blue components. In this way, the user is allowed to select a small subset of colors, called a palette, from the full range of $2^{24} = 16,777,216$ colors. The drawback of this scheme is that it restricts the number of colors which may be simultaneously displayed.

This paper addresses the problem of displaying color images using a restricted palette. Since natural images typically contain a large number of distinguishable colors, displaying such images with a limited palette is difficult. The approach we take is structured around algorithms for performing two tasks. The first task, called color palette design, selects the best possible set of colors for a particular image. The second task, called pixel mapping, associates each pixel of the image with a color from this palette to yield the highest quality image.

A number of approaches [1, 2] have been suggested for the design of color palettes which involve iterative refinement of some initial palette. However, these algorithms suffer from the disadvantage of being computationally intensive. In many applications, the efficiency of algorithms for performing color palette design and pixel mapping is very important. This is particularly true in applications involving large image data bases. In these applications, images are in some common format which must be displayed on monitors with different limitations on color palette size. In such an environment, the color palette design and the pixel mapping functions must be performed at the time an image is displayed. This makes computational efficiency of critical importance.

Even if the color palette is transmitted with each image, standard image coding techniques require that the pixel mapping be performed locally at the time each image is displayed. This is because standard image coding techniques make use of the high correlation between the values of the primary colors of neighboring pixels to reduce transmission bit-rate. The color palette indices of neighboring pixels are not highly correlated and, therefore, cannot be coded with high efficiency. Thus, even if a color palette of correct size is provided with each stored image, the pixel mapping function must be performed after the image is retrieved from the data base and decoded into a full color format.

The first problem we address is the problem of designing color palettes. In general, a color palette will accurately display a particular image if the true color of each pixel in the image can be well approximated by some color from the palette. Heckbert [2] proposed a method for constructing an initial color palette by assigning to each color from the palette an approximately equal number of pixels in the image. This method is based on the construction of a binary tree in which each leaf contains an approximately equal number of similar pixel colors which can be described by a single quantization color. The tree is constructed by recursively determining the leaf node with the largest number of pixels and splitting it into two new nodes with equal numbers of pixels. To improve this initial color palette, Heckbert proposed an iterative procedure, derived from the Linde, Buzo, Gray (LBG) algorithm [3] for quantizer design, which attempts to minimize the total squared error (TSE) between the original pixels and colors assigned from the palette. Later, Braudaway [1] proposed constructing an initial color palette by centering colors at peaks in the color-value histogram. To avoid concentrating too many colors about one peak, the histogram values are reduced in a region surrounding each assigned color after the color is selected. Beginning with this initial palette, Braudaway also used the iterative LGB algorithm as suggested by Heckbert. An approach similar to that of Braudaway's has also been used by Gentile, Allebach and Walowit [4] to study the effects of color transformations on the quality of quantized images.

In both Heckbert's and Braudaway's methods, the initial condition is chosen to allocate more quantization levels to regions of the color space with larger numbers of pixels. However, neither initial condition attempts to minimize the TSE or any other objective measure of error. Therefore, the LBG algorithm is used to improve these initial color palettes by searching for a local minimum of the TSE. While the LBG algorithm does reduce the initial TSE, it is greatly influenced by these initial palettes. Consequently, the local minimum which is reached is unlikely to be close to the global minimum. In addition, the LBG algorithm is very computationally intensive since, at each iteration, it requires a search of the full color palette for each pixel. Also, the LBG algorithm destroys the tree structure of Heckbert's initial palette. This is important since the tree structure could otherwise be used to reduce the computation required for palette searches.

This paper describes a hierarchical approach to the design of color palettes for natural images. The proposed algorithms attempt to produce color palettes which minimized an objective error criteria. Therefore, they result in high quality displayed images with minimum artifacts [7, 8]. We use vector quantization (VQ) [9] as a basis for addressing the design of good color palettes. In this framework, the colors in the palette can be considered as codes in a code book, and the pixel mapping as the image pixel coding. In addition, our approach emphasizes the use of color palettes with a binary tree structure. This tree structure has the advantage of substantially reducing the computation required both to design a palette and to quantize colors using a palette. We develop a basic binary splitting approach which creates tree structured color palettes using a minimum TSE criteria. This allows for very efficient pixel mapping by making search times proportional to the tree depth.

Empirical evidence indicates that the best color palettes are not necessarily ones which minimize TSE. This is because the subjective error resulting from quantizing a pixel's color value can not be treated independently of the quantization of spatially adjacent pixels. Standard VQ based algorithms do not address many of these subjective qualities because they generate a codebook based on the distribution of pixel colors without consideration of their spatial relationships. We develop two approaches to modifying the basic binary splitting algorithm to account for these subjective qualities. The first approach computes a weighted TSE (WTSE) by applying a subjective weighting to the squared error at each pixel and uses an extension of the basic binary splitting VQ algorithm to minimize this WTSE. The subjective weighting reflects variations in our ability to perceive small intensity changes as a function of the spatial intensity gradient at each pixel. The second approach modifies the order in which the binary splitting algorithm splits nodes so as to reduce the correlation of quantization errors between adjacent pixels. This approach is based on reducing the sizes of interiors of constant color areas in the displayed image.

The second part of this paper addresses the problem of optimum pixel mapping once a color palette has been selected. Pixel mapping involves assigning a color from the palette to each pixel in the image in a way which produces the highest quality display. Although the TSE can be minimized by assigning each pixel to the color that is closest to its original color, higher quality displays are often produced by using dithering methods to assign colors. These methods attempt to match the local color averages of the displayed image to the local color averages of the original image. A variety of dithering techniques have been investigated for displaying bi-level or half-toned images (i.e. a color palette containing only the colors black and white). Both Heckbert and Braudaway suggest the use of a dithering technique known as error diffusion [11] which was originally proposed for the display of bi-level images but is easily extended to multilevel quantization. Error diffusion can significantly improve the quality of many displayed images when the palette size is small. However, when used in conjunction with an unstructured color palette, error diffusion requires a search of the full color palette for each pixel of the image and is therefore quite computationally intensive.

We investigate a number of approaches for improving display quality given a fixed color palette. These methods produce more accurate average color at the cost of added noise and increased computation. By using error diffusion in conjunction with color palettes having a binary tree structure, we substantially reduce the computation required to quantize an image. In addition, we propose a modified algorithm for performing the error diffusion. This algorithm avoids nonlinear oscillations which can occur due to the nonuniform quantization of the color space. This results in better image reproduction with fewer artifacts.

The paper is organized as follows. Section 2 starts by defining the color space coordinate system in which the palette design algorithms operate. Section 2.1 describes the basic binary splitting approach to the design of color palettes using a minimum TSE criteria. Sections 2.2 and 2.3 develop extensions to the basic binary splitting algorithm which account for subjective measures of image quality. Section 3 describes both ordered dither methods and methods for efficiently implementing error diffusion techniques using color palettes having a binary tree structure. Finally, section 4 reports on simulation results of the various techniques.

2 Color Palette Design

We first define some notation. The image is assumed to be on a rectangular grid of N pixels. The set of all grid points is denoted by S, and its members $s \in S$ may be explicitly written as $s = (i, j)$ where i is the row index and j is the column index. The color value of the pixel at grid point s is denoted $\mathbf{x}_s = [r_s, g_s, b_s]^t$ where the components are the red, green and blue tristimulus values for the pixel [13] and superscript t means transpose.

For a typical monitor, there is a nonlinear relationship between the input value of the primary and its displayed intensity. For a particular primary such as red, this relationship may be approximated by

$$I_r = r^\gamma$$

where I_r is the intensity, r is the input to the display and $\gamma \approx 2.3$. More accurate models for this relationship generally include constant, linear and logarithmic terms [4]. In order to compensate for this nonlinear relationship between input and luminance, it is common for video cameras and other image capture devices to predistort their output by the inverse relationship.

$$r = I_r^{(1/\gamma)}$$

We shall refer to the predistorted primary color values as the *gamma corrected* colors. In effect, this is the coordinate system being used if one manipulates color image data and directly displays it without correction. It is important to emphasize that the gamma corrected colors are *not* linear in intensity, and that the gamma correction, and therefore the resulting coordinate system, varies with changes in a display's parameters. However, the gamma corrected colors have the advantage that they may be directly displayed without transformation, and that they are less susceptible to noise than the corresponding linear primary intensities [13].

Alternatively, standard coordinate systems have been defined for the representation of colors. Two such coordinate systems are the $L^*a^*b^*$ and the $L^*u^*v^*$ color spaces [12]. These coordinate systems are nonlinearly related to the primary intensities, and are chosen so that a fixed euclidean distance represents a fixed perceptual distance independent of position in the space. Gentile, Allebach and Walowit have studied the application of the $L^*u^*v^*$ coordinate system to the problems of image quantization [4], halftoning [5] and gamut mismatch [6]. This work indicates that these coordinate systems can substantially improve quantized image quality relative to standard coordinate systems which are linearly related to primary luminances. Therefore, it is reasonable to expect that perceptually based coordinate systems such as $L^*u^*v^*$ could improve the quality of displayed images relative to gamma corrected coordinates. However, it will be assumed throughout this paper that the original image is specified in terms of gamma corrected coordinates, since this avoids the computational requirements of coordinate transformation methods. All, the techniques developed can of course be applied using perceptually based color spaces.

2.1 Binary Tree Palette Design

The objective of the palette design algorithm is to partition S into M disjoint sets or clusters where M is the predetermined palette size determined by hardware constraints. The algorithm proposed here constrains the partitioning of S to have the structure of a binary tree. Each node of the tree represents a subset of S, and the children of any node partition the members of the parent node into two sets. The set of image pixels corresponding to node n is denoted C_n. For purposes of illustrating the operations of the algorithm, we will number the nodes so that the root is 1 and the children of node n are $2n$ and $2n+1$. Since the leaves of the tree form a partition of S, the pixels in S may be coded by representing all pixels in each leaf node with a single color in the palette. Only the members of the tree's leaf nodes need be stored when the algorithm is implemented. Each of the leaf node sets can be stored as a linked list. When a node is split, the linked list may be reorganized into two lists without relocating any of the image color data.

The method for generating the binary tree is specified by the number of leaves, M, the method of splitting a node into its two children, and the order in which the nodes are split. The methods used for splitting nodes and determining which nodes to split both attempt to minimize the total squared difference between the actual and quantized image. The TSE is defined by

$$TSE = \sum_{\substack{\text{all leaves} \\ n}} \sum_{s \in C_n} ||\mathbf{x}_s - q_n||^2$$

where q_n is the quantization value used to represent the colors in the set C_n. In order to restrict the complexity of the algorithm, second order statistical properties will be used to determine the order and manner in which nodes will be split. The three required cluster statistics are

$$\begin{aligned} R_n &= \sum_{s \in C_n} \mathbf{x}_s \mathbf{x}_s^t \qquad (1) \\ m_n &= \sum_{s \in C_n} \mathbf{x}_s \\ N_n &= |C_n| \end{aligned}$$

where R_n is a 3x3 matrix, and m_n is a 3 component vector. Since the mean value of a cluster is the point from which there is minimum squared deviation, the quantization value of a cluster, q_n, is assumed equal to the cluster mean.

$$q_n = \frac{m_n}{N_n} \ .$$

We also define the cluster covariance as

$$\tilde{R}_n = R_n - \frac{1}{N_n} m_n m_n^t \ .$$

The splitting of a node into two nodes is equivalent to choosing two new quantization levels, q_{2n} q_{2n+1}, and associating each member of the cluster with the closer quantization level. This, in turn, is equivalent to selecting a plane which best splits the cluster's colors. In the proposed algorithm, we determine the direction in which the cluster variation is greatest, and then split the cluster with a plane which is perpendicular to that direction and passes through the cluster mean. For a large cluster with Gaussian distribution it can be shown that this strategy is optimal.

More specifically, we determine the unit vector e which maximizes the expression

$$\sum_{s \in C_n} ((\mathbf{x}_s - q_n)^t e)^2 = e^t \tilde{R}_n e \ .$$

Since $\tilde{R}$ is symmetric, the solution is the eigenvector, e_n, corresponding to the largest or principal eigenvalue λ_n of $\tilde{R}$. The total squared variation in the direction e_n is therefore

$$\sum_{s \in C_n} ((\mathbf{x}_s - q_n)^t e_n)^2 = \lambda_n \ . \qquad (2)$$

Once the principal eigenvector has been determined, points in C_n can be sorted into two sets C_{2n} and C_{2n+1} in the following way.

$$\begin{aligned} C_{2n} &= \{s \in C_n : e_n^t \mathbf{x}_s \leq e_n^t q_n\} \qquad (3) \\ C_{2n+1} &= \{s \in C_n : e_n^t \mathbf{x}_s > e_n^t q_n\} \ . \end{aligned}$$

Finally, the new statistics for each node may be calculated by first calculating R_{2n}, m_{2n} and N_{2n} for node $2n$, and then applying the relations

$$\begin{aligned} R_{2n+1} &= R_n - R_{2n} \\ m_{2n+1} &= m_n - m_{2n} \\ N_{2n+1} &= N_n - N_{2n} \ . \end{aligned} \tag{4}$$

The order in which nodes are split is chosen with the objective of producing the greatest reduction in TSE at each stage of the algorithm. Because this strategy does not look ahead to the results of further splits, it is not necessarily optimal; however one would expect it to perform well in most practical situations. When a node is split its variance is mainly reduced along the direction of the principal eigenvector. Therefore, it is reasonable to assume that the reduction in TSE should be proportional to the total squared variation along the direction of the principal eigenvector, e_n, of $\tilde{R}_n$. For this reason, the principal eigenvalue, λ_n is used as a measure of the expected reduction in TSE if node n is split. Given this approximation, the best allocation of a single quantization level is to split the leaf with the largest principal eigenvector.

The basic binary quantization algorithm may now be described.

1. Set $C_1 = S$.

2. Calculate R_1, m_1 and N_1.

3. Do the following $M - 1$ times

   3.1 Find the leaf, n, such that λ_n is largest.

   3.2 Use (3) to form the new nodes $2n$ and $2n + 1$.

   3.3 Calculate R, m and N for the new nodes using (4).

For typical image sizes (e.g. 512x512 pixels) the computation is dominated by steps which must be performed at each pixel of the image. These steps consist of the splitting of clusters using (3) and the computation of statistics using (4) and (1). Assuming an approximately balanced tree, each pixel of the image is involved in $\log_2 M$ splits, each requiring 3 multiplications to compute (3). The initial computation of the matrix R_1 requires 6 multiplies per pixel due to the matrix's symmetry. Assuming approximately balanced splitting of each node, $3N \log_2 M$ multiplies are required to compute the remaining values of R_n. This results is a total of $6N(\log_2 M\ + 1)$ multiplications and slightly fewer additions for the design of the color palette. Therefore, for $M > 2$, fewer multiplications are required for the design of the complete color palette than a single iteration of the LBG procedure described in section 2.4.

2.2 Subjectively Weighted TSE Criteria

In this section, we develop a framework for incorporating subjective measures of quality through the use of a weighting function which accounts for local properties of the image. This weighting can be applied using the same framework of the basic splitting algorithm of section 2.1, with the substitution of a weighted total square error (WTSE) in place of the true mean square error. The WTSE is defined as

$$WTSE = \sum_{\substack{\text{all leaves} \\ n}} \sum_{s \in C_n} W_s(\mathbf{x}) ||\mathbf{x}_s - q_n||^2$$

where W_s is a function of $\mathbf{x}$ in some neighborhood of s. Regions of the image which are subjectively more sensitive to errors in quantization are given a larger weight. By applying the same arguments used in deriving the basic splitting algorithm, we calculate the following forms for the new cluster statistics.

$$\begin{aligned}
R_n &= \sum_{s \in C_n} W_s \mathbf{x}_s \mathbf{x}_s^t && (5) \\
m_n &= \sum_{s \in C_n} W_s \mathbf{x}_s \\
N_n &= \sum_{s \in C_n} W_s \\
\tilde{R}_n &= R_n - \frac{1}{N_n} m_n m_n^t \\
q_n &= \frac{m_n}{N_n}
\end{aligned}$$

We first consider some properties that the weight W_s should possess. For simplicity, W_s will only depend on the gamma corrected luminance component, y_s, of the image. In the NTSC system, y_s is given by

$$y_s = [0.300, 0.586, 0.115]\mathbf{x}_s \ . \tag{6}$$

If I_s is the true luminance at position s, then y_s is approximately given by $y_s \approx I_s^{1/\gamma}$. Weber's law [14] suggests that any measure of subjective image quality should be invariant to linear scaling of the color component intensities. However, this is not true of the TSE based on gamma corrected color components. Scaling I_s by the constant c^γ results in y being scaled by c and the TSE being scaled by c^2. Therefore, to ensure the invariance of the WTSE to scaling of I_s, W_s should have the property that

$$W_s(c\mathbf{x}) = \frac{1}{c^2} W_s(\mathbf{x}) \ . \tag{7}$$

This suggests that W_s should be inversely proportional to a quadratic function of a local linear function.

The most prominent artifact of a limited palette is false contouring. False contours result from mapping smoothly varying regions of an image to a small number of colors from the color map. Instead of displaying slow variations of color across the region, smaller regions of constant color are displayed with abrupt color changes across region boundaries. These abrupt color changes are perceived as contours in the displayed image.

A reasonable measure of the visibility of a false contour is the width of uniformly quantized region. Since the luminance component has the greatest variation, we will use it to estimate the likely size of false contours. Figure 1 is a schematic representation of a one-dimensional slice through the luminance component of an image along the direction of the gradient. The two curves are the true luminance and the quantized luminance. Notice that the width of the uniformly quantized region, Δ, is equal to the quantization step size, E,

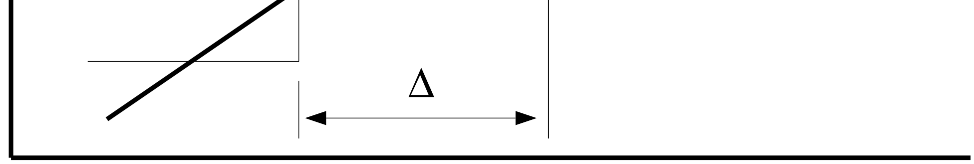

Figure 1: This figure illustrates the relationship among the gradient of the image luminance ($\bigtriangledown y$), the size of quantization steps (E) and the width of regions with fixed quantization value (Δ).

divided by the gradient size. This suggests that for a region of the image with uniform quantization

$$\sum_{s \in C_n} \frac{||\mathbf{x}_s - q_n||^2}{|| \bigtriangledown y_s||^2} \approx (\text{area}) \frac{E^2}{12|| \bigtriangledown y_s||^2} \approx (\text{area}) \frac{\Delta^2}{12} \ .$$

In practice, the gradient of y_s is not constant, so a more accurate estimate may be obtained by convolving the magnitude of the gradient with a smoothing kernel h_s. The form of W_s which is used is

$$W_s = \left(\frac{1}{h * (\min\{|| \bigtriangledown y||, 16\} + 2)} \right)^2 \tag{8}$$

where y takes on values between 0 and 255, and h is a kernel which filters the gradient estimates. The effect of $\bigtriangledown y$ is limited to the range between 2 and 16 since values below 2 are visually indistinguishable, and values greater than 16 correspond primarily to edges. Also, these limits allow fixed-point implementation of the algorithm by limiting dynamic range.

The increase in computation due to the use of the WTSE criteria comes from the additional multiplications required by the appearance of W_s in (5) and the computation of the weightings in (8). (5) requires an additional $1.5N(\log_2 M \ + 1)$ multiplications, increasing the complexity of the basic algorithm by 25%. The major computational cost of calculating the weights is the computation of the gradient, the convolution with h and the squaring of the weights. The division can be avoided by using power of two quantization. The computation required for filtering depends on the filter size, and weights. The gradient calculation together with the implementation of a 5x5 filter of 1's and the squaring of the result requires $13N$ arithmetic operations including $1N$ multiplications. In practice, the calculation of these weights adds substantially to the total computation time.

2.3 Erosion-Based Weighting

Another method for improving the quality of the quantization is to develop an error criteria which depends directly on the qualities of the quantized image. Such a method can be used to dynamically focus quantization values in regions of the quantized image which are problematic. More specifically, if regions of the quantized image that may result in false contouring can be efficiently identified while the binary tree is being created, then the relevant clusters can be split to avoid artifacts.

The visual effect of false contouring is caused by the highly correlated nature of the error which results from large regions being quantized with a single value. Even if the TSE (or WTSE) is small in such regions the errors will be visually significant because they will form well defined patterns. However, regions of the image which are quantized with single values may be identified by searching the two dimensional field of quantization code words for regions with large interiors. The interiors of each region formed by a fixed code word value may be measured using the technique of erosion from morphological filtering [15]. This is done by selecting a fixed pattern of lattice points and counting the number of times the pattern may be uniquely positioned completely within a region consisting of a single code word value. This operation is shown graphically in Figure 2. If a square block of nine points is used as the eroding set, then this is equivalent to counting the number of pixels in a cluster for which all eight neighbors of that pixel belong to the same cluster. For this reason, the interior size ω_n of cluster n is a measure of the number of interior pixels in the cluster.

Figure 2: Sample erosion with nine point block.

After an initial allocation of M_0 code words, the weights ω_n may be used to bias the order in which clusters are split by more heavily weighting principal eigenvalues of clusters with large interiors. Instead of choosing the cluster with the largest eigenvalue, λ_n, the weighted eigenvalue, $\omega_n \lambda_n$, is used. All experimental results use $M_0 = (2/3)M$. We have found that for palettes with 256 colors this value of M_0 allocates a sufficient number of colors to eliminate regions with large interiors.

The binary quantization algorithm with erosion-based weighting may now be described.

1 Set $C_1 = S$.

2 Calculate R_1, m_1 and N_1.

3 Do the following $M_0 - 1$ times.

3.1 Find the leaf, n, such that λ_n is largest.

3.2 Use (3) to form the new nodes $2n$ and $2n + 1$.

3.3 Calculate R, m and N for the new nodes using (4).

4 Calculate the interior weights, ω_n, for all leaf nodes, n.

5 Do the following M-M_0 times.

5.1 Find the leaf, n, such that $\omega_n \lambda_n$ is largest.

5.2 Use (3) to form the new nodes $2n$ and $2n+1$.

5.3 Calculate R, m and N for the new nodes using (4).

5.4 Compute ω_{2n} and ω_{2n+1} for the two new leaves.

The calculation of the weights ω_n represent only a modest overhead in computation because they can be efficiently computed using only compare and logical operations. Step 4 requires that an interior weight be computed for each region of the image. For a K point eroding set, the worst case number of comparisons required for step 4 is $N(K-1)$. However, this is a conservative estimate since the search for differing quantization levels can be performed sequentially, and can be stopped when unlike quantization levels are found. Tests using a 9 point eroding set for quantizing a large sample of images have found that an average of $1.4N$ comparisons are needed for computing the interior weights. The total computation required for all applications of step 5.4 is generally less than is required for step 4. This is because the splitting of a cluster only effects the interior points of the cluster being split. Therefore, time required to recompute these two weights is proportional to the number of pixels in the cluster being split. Since the number of pixels contained in the last $M/3$ of clusters split is generally much less then N, this represents a small amount of computation.

2.4 The LBG Algorithm

The LBG Algorithm [3, 9] is a method used in VQ for iteratively refining a code book. The algorithm is quite general, and has also been applied to wide variety of problems in pattern recognition and clustering [10]. Since the problem of designing an optimal code book is equivalent to finding a set of palette colors, $\{q_n : 1 \leq n \leq M\}$, which best approximate the set of image pixel colors, $\{\mathbf{x}_n : 1 \leq n \leq N\}$, the LBG algorithm is directly applicable to the color quantization problem. In the case of minimum TSE palette design, the algorithm is defined as follows:

1. Choose an initial color palette, $\{q_n : 1 \leq n \leq M\}$.

2. Form M pixel clusters, $C_m = \{\mathbf{x}_n : ||\mathbf{x}_n - q_m|| \leq ||\mathbf{x}_n - q_k||, 1 \leq k \leq M\}$

3. Recompute the color palette using, $q_m = \frac{1}{|C_m|} \sum_{s \in C_n} \mathbf{x}_s$

4. Repeat 2 and 3 until no more changes occur or the decrease in TSE is less than a predetermined amount.

Each step of the LBG algorithm causes the TSE to either decrease or remain the same. Therefore, the algorithm is guaranteed to converge to a minimum of the TSE. However, this minimum is generally not the global minimum of the TSE. Therefore, the initial palette plays a key role in determining the local minimum to which the algorithm converges. In fact, the experimental results of Section 4 indicate that the LBG algorithm will improve initial color palettes of poorly quantized images, but it will not significantly change palettes of well quantized images. Consequently, we take the view that the LBG algorithm is best used as a post-processing algorithm for improving suboptimal initial palettes.

A significant disadvantage of the LBG algorithm is its computational cost. The forming of clusters in step 2 generally requires NM distance calculations, and this step must be performed in each iteration. Therefore, one iteration of the LBG algorithm requires $3NM$ multiplications. Finally, we noted that binary tree structure of a palette is destroyed by the LBG algorithm. Without the tree structure, pixel quantization requires a full search of the palette, or NM distance calculations. However, pixel quantization with the tree structure requires only $N \log_2 M$ inner products and compares. This is particularly important in the error diffusion applications discussed below.

3 Pixel Mapping

Once a color palette has been selected, the TSE of the displayed image is minimized by mapping each pixel to the color from the palette which is closest to its original color. The pixel mapping techniques described in this section attempt to improve the quality of the displayed image by accepting a larger TSE so that the local average color of the displayed and original images may be matched. These techniques reduce error power at low spatial frequencies and increase error power at high spatial frequencies where error is less noticeable. This section discusses two types of pixel mapping techniques, ordered dither and error diffusion, which have been used widely for the halftoned display of grey-scale images. The application of these techniques to the display of color images with limited color palettes raise difficulties which do not exist in the display of bi-level images.

3.1 Ordered dither

Ordered dithering is a method for improving the appearance of a displayed image by adding a pseudorandom noise pattern, $d_{(k,l)}$, to blocks of pixels, x_s, before quantizing. Therefore, the image before quantization, $\tilde{x}$, is given by

$$\tilde{x}_{(i,j)} = d_{(i \bmod n, j \bmod n)} + x_{(i,j)},$$

where $d_{(k,l)}$ is a n by n matrix of dithering values. $d_{(k,l)}$ is chosen to have an average value of zero and an energy spectrum with minimum energy at low spatial frequencies. The amplitude of the noise pattern is chosen so that any area of constant color value is quantized into a variety of nearby color values, thereby breaking up regions of false contouring. This reduces the correlated error patterns that are characteristic of areas of false contouring.

The application of ordered dithering to the display of color images is complicated by the nonuniform spread of colors in the palette. In image-specific color palettes, the distance between nearby colors varies significantly across the color space. Therefore, it may be impossible to determine an amplitude which will sufficiently dither all areas of constant color value in the image without adding noticeable noise to other areas. One method of addressing this problem is to adjust the dithering to the distance and orientation between nearby colors in the palette. This can be done by first determining two candidate quantization colors, q_1 and q_2, which are nearest to the actual pixel color. The noise pattern value for the pixel can then be scaled by the distance between quantization colors and rotated in color space by the

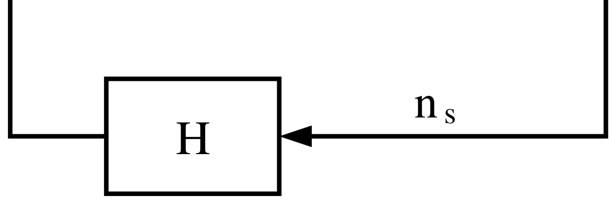

Figure 3: Block diagram of error diffusion filter. The quantization error is propagated forward to reduce the low frequency component of the error.

orientation of the colors to form the dithered signal

$$\tilde{x}_{(i,j)} = (q_2 - q_1)d_{(i \bmod n, j \bmod n)}, + x_{(i,j)} \tag{9}$$

where d takes on scalar values on the interval $[-1/2, 1/2]$. The dithered value can then be quantized to one of the two levels.

This approach adjusts the dithering to the non-uniform spread of quantization colors; however, it has a number of disadvantages. In regions which change significantly enough to cause q_1 and q_2 to change, the dither noise will be modulated by the changing quantization colors. This will cause the spectrum of the dither noise to spread to low spatial frequencies. Because the dither is only added along the direction of the differing quantization values, the average error in directions orthogonal to that component will not be effected. The computational cost of searching for candidate quantization colors can be reduced by restricting the search to colors which appear in a limited neighborhood of the pixel; however, this search still possesses a substantial computational overhead. Finally, no ordered dithering method guarantees that after quantization the error spectrum will continue to be concentrated at high spatial frequencies.

3.2 Error diffusion

The object of error diffusion is to quantize the image in such a way that the average value of the quantized image is the same as the average value of the true image [11, 16]. Figure 3 illustrates the method. The pixels are chosen in some ordering (raster ordering is commonly used), and the residual quantization error is propagated forward to unquantized pixels. The error is propagated by computing an adjusted value of the pixel color, $\tilde{x}_s$, formed by summing the actual color with a weighted sum of previous errors. The equations for error diffusion are as follows

$$\begin{aligned} \tilde{x}_s &= x_s + \sum_{k<s} h_{s-k} n_k \\ q_s &= Q(\tilde{x}_s) \\ n_s &= \tilde{x}_s - q_s \ , \end{aligned} \tag{10}$$

where $s - k = (s_1 - k_1, s_2 - k_2)$ and $s < k$ means $(s_1 = k_1 \ \& \ s_2 < k_2)$ or $s_1 < k_1$. We note that while these equations have a somewhat different form then those originally presented by Floyd and Steinberg, they are equivalent. Since x_s is a vector, (10) describes a system of vector equations where h_s is a matrix for each value of s. For simplicity, the h is usually chosen to be diagonal. The two-dimensional Fourier transform of h is given by

$$H(\omega) = \sum_{s>0} h_s e^{-j\omega\cdot s},$$

where $\omega = (\omega_1, \omega_2)$ and $\cdot$ is the notation for inner product.

Notice that the variable, n_s, is the error generated by the quantizer, but it is not the difference between the actual and quantized image. The actual displayed image error and its relationship to the quantizer error is given by

$$\begin{aligned} \varepsilon_s &= x_s - q_s \\ &= x_s - (\tilde{x}_s - n_s) \\ &= n_s - \sum_{k<s} h_{s-k} n_k \ . \end{aligned}$$

The two-dimensional Fourier transform then yields

$$\mathcal{E}(\omega) = (I - H(\omega))\, N(\omega) \ .$$

Therefore, we see that the spectral components of the displayed error may be shaped by choosing the filter $I - H(\omega)$ correctly. In general, this filter should have a high-pass characteristic since it is desirable to suppress the low-frequency error components. Billotet-Hoffmann and Bryngdahl have suggested a simple filter kernel which we have used in all experimentation [16].

The methods of color palette generation described in the previous sections all generate a color palette and a mapping from the image pixels to that palette. However, in general, palette design and pixel quantization can be performed separately. In the case of minimum TSE coding using an unstructured color palette, quantization is performed by

$$Q(x) = \arg\min_{q_n}\{||x - q_n||^2\} \ .$$

This is very computationally intensive since quantizing an image requires NM distance calculations, each of which requires three multiplies, three subtractions and a comparison. In the case of a palette with binary tree structure, an approximate minimum TSE coding, $Q_b(x)$, can be performed with order $N \log M$ computation. This is done by storing the principal eigenvectors and cluster means of all the internal tree nodes during the tree construction. These values can then be used to uniquely descend through the tree using the rule from (3). Notice, that both types of quantization can be performed for colors outside the original image. This is important since error diffusion will require the quantization of colors not in the original image. However, this leads to a unique problem associated with the application of error diffusion to image specific palettes.

The advantage of an image-dependent palette is that quantization values are densely spaced only in regions of the color space with significant numbers of pixels. This advantage can be quite significant because for typical images a large amount of the color space is empty. The situation may be visualized by considering a region of an image in which the color varies smoothly (such a region is more likely to exhibit the problems typical of a limited palette.) The image is then a smooth mapping from a two-dimensional surface into a three-dimensional space. The range of that mapping is, in general, a manifold with dimension ≤ 2. In particular, the local dimension of the manifold is given by the rank of the matrix

$$\begin{bmatrix} \frac{\partial r}{\partial i} & \frac{\partial g}{\partial i} & \frac{\partial b}{\partial i} \\ \\ \frac{\partial r}{\partial j} & \frac{\partial g}{\partial j} & \frac{\partial b}{\partial j} \end{bmatrix}$$

where i and j are temporarily considered to take values on the real line. Most often this matrix will have rank 2 since the color components are likely to vary with some degree of independence. However, the manifold will have dimension 1 if two of the components are a function of the third. This can occur if the region has fixed hue and saturation and varies only in intensity. The most degenerate case is when the color of an region is fixed and the manifold is zero-dimensional.

When the color palette is designed, the majority of colors are allocated around these surfaces in the color space. Unfortunately, the sparse nature of these surfaces can cause problems when colors outside the set are quantized. Error diffusion accumulates errors made in the pixel coding until those errors are negated by errors of opposite polarity. However, if all the quantization levels fall to one side of the manifold then there will be no local colors to offset accumulated error. This situation can occur if the manifold is a convex surface since the minimum TSE criteria will tend to allocate color on the concave side of the manifold. Ultimately, the error will become so great that either the color will become saturated to some extreme value, or more likely, a color from another manifold (region of the image) will be chosen. In this situation the assumption of independence between the input to the quantizer and the quantization error becomes invalid, and the nonlinear feedback structure of error diffusion can lead to a unstable condition in which large oscillations in quantization error occur between colors far from the original color. This, of course, creates noticeable artifacts in the resulting image.

We would like to constrain the filter, H in figure 3, to operate in a region in which the

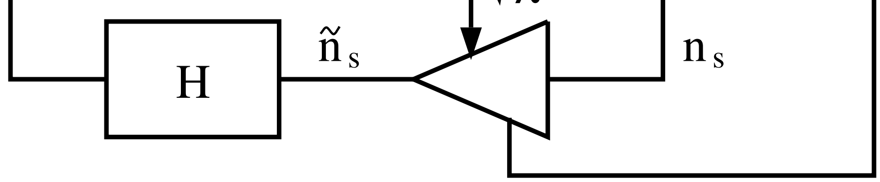

Figure 4: Block diagram of modified error diffusion filter. The feedback is clipped if magnitude of the error is significantly larger than the longest axis of the corresponding cluster formed in the palette design process.

quantizer error, n_s, is of the order of the distance between quantization colors within the manifold. A reasonable method for detecting when the filter is oscillating is to compare the error magnitude to the diameter of the cluster associated with the quantization color. This can be done by storing along with the quantization color the associated principle eigenvalue of the cluster. Since the principle eigenvalue of the cluster is proportional to the square of the cluster's diameter along its longest axis, we can determine when oscillations occur by comparing the error magnitude to $\sqrt{\lambda}\alpha$ where α is a constant. When oscillations are detected, the filter can then be damped by opening the feedback loop, and the unadjusted pixel intensities, x_s, can be quantized instead of the highly biased values, $\tilde{x}_s$. This combined operation is simply performed by "clipping" the input of the filter to a magnitude of $\sqrt{\lambda}\alpha$.

$$\tilde{n}_s = \begin{cases} n_s & \text{if } |n_s|^2 < \lambda\alpha^2 \\ 0 & \text{otherwise} \end{cases}$$

A diagram of the modified error diffusion process is shown in Figure 4.

4 Results

The previous sections have proposed a number of algorithms, each of which attempts to maximize performance by minimizing a measure of displayed image distortion (e.g. TSE, WTSE and region interior size). This section compares these algorithms by measuring their performance with respect to all of the quality measures which have been discussed. Besides providing a means for comparing algorithms, the results of this section offer insights into the validity and significance of the various performance criteria which have been proposed.

The algorithms were tested on a variety of images available on public image databases, and four images were selected for comparisons. Figure 5 shows monochrome versions of the four color images used. Images 1 and 2, known as Lenna and Tiffany respectively, are characterized by large smooth regions of light skin tones which are particularly difficult to quantize transparently. This is probably due to the combined effects of a sensitive region of luminance/hue together with strong prior expectations on the part of the viewer. Image 1 contains colors from a larger area of the color space than image 2, and it suffers from the most noticeable false contouring of the four images used for comparisons. Image 3, known as F16, is relatively easy to quantize transparently because its large saturated white background contains colors concentrated in a small region of the color space. Image 4, known as Peppers, contains colors from many disjoint regions of the color space and produces significantly higher TSE than the other images.

Table 1 compares five color quantization algorithms with respect to three performance criteria. The comparisons are shown for images 1 and 4 which are the most challenging images to quantize. The first algorithm, labeled (8,8,4), uses a image independent separable color map with 8 red, 8 green, and 4 blue quantization levels. The second algorithm, labeled HIST, is a generic histogram-based method similar to that used for the initialization step in [1]. The final three algorithms, labeled BS, WBS, and EBBS, are the basic binary splitting algorithm of Section 2.1, the weighted binary splitting algorithm of Section 2.2, and the erosion-based binary splitting algorithm of Section 2.3. The WBS method uses a 5x5 moving

Lenna; Tiffany
F16; Peppers

a	b
c	d

Figure 5: Monochrome versions of color images used in comparisons of algorithms. The images are referred to as a) Lenna; b) Tiffany; c) F16; d) Peppers.

average filter to smooth the gradient components, and the EBBS uses the nine point block of Figure 2. Three criteria are used to judge algorithm performance: the RMS error given by $RMSE = \sqrt{\frac{TSE}{N}}$, the WRMS error given by $WRMSE = \sqrt{\frac{WTSE}{N}}$, and the average codeword interior size (ACIS) computed with the nine point eroding block.

	Image 1			Image 4		
Algorithm	RMSE	WRMSE	ACIS	RMSE	WRMSE	ACIS
(8,8,4)	23.66	23.35	187	22.30	21.92	189
HIST	7.87	7.36	28	11.39	10.22	31
BS	5.96	5.28	5.7	8.26	7.28	6.0
WBS	6.31	5.08	3.4	8.73	7.06	3.2
EBBS	6.10	5.27	2.3	8.43	7.27	1.5

Table 1: Color quantization algorithms compared according to RMSE, WRMSE, and average codeword interior size (ACIS).

Several observations can be made from Table 1. It is clear that the algorithms generating image dependent color maps yield significantly better performance than the fixed (8,8,4) color map according to all performance criteria. The generic histogram algorithm, which is not developed from maximizing any performance criterion, is outperformed by the three tree structured algorithms according to all criteria. It should be noted that the displayed images produced by the first two algorithms show noticeable false contouring and are clearly of lower subjective quality than those produced by the final three algorithms. The differences between the tree-structured algorithms are less obvious and, in many cases, can be identified only under close examination.

It is not surprising that, for each of the three criteria, the algorithm giving the best performance is the one that was developed to maximize performance relative to that criteria. However, some insight into the significance of the three criteria can be gain by comparing the results of the three algorithms. The BS and EBBS algorithms differ only in the selection of codewords which are split by the last 25% of the node splitting operations. Although the EBBS gives a 2% increase in the RMSE by using the erosion-based criterion for selecting codewords to split, it achieves greater than 50% reduction in ACIS, thus suggesting that the RMSE criterion is a poor measure of false contouring in the displayed image. The EBBS algorithm performs roughly equal to the BS algorithm relative to the WRMSE criterion, reflecting the fact that codewords with large interiors are often found in regions which are given high weights by equation (8). The relationship between the weighting function of Section 2.2 and the erosion-based weighting of Section 2.3 is also reflected in the performance of the WBS algorithm. Although it does not directly attempt to reduce interior size, Table 1 shows that the WBS algorithm produces significantly smaller ACIS compared with the basic BS algorithm. Both the WBS and EBBS algorithms produce noticeably less false contouring than the BS algorithm.

The LBG algorithm described in Section 2.4 can be applied to improve the color maps generated by each of the algorithms considered in Table 1, and Figure 6 shows the RMSE that results at each iteration. Since the first two algorithms were not designed to minimize any error function, the early LBG iterations offer significant reduction in RMSE for these

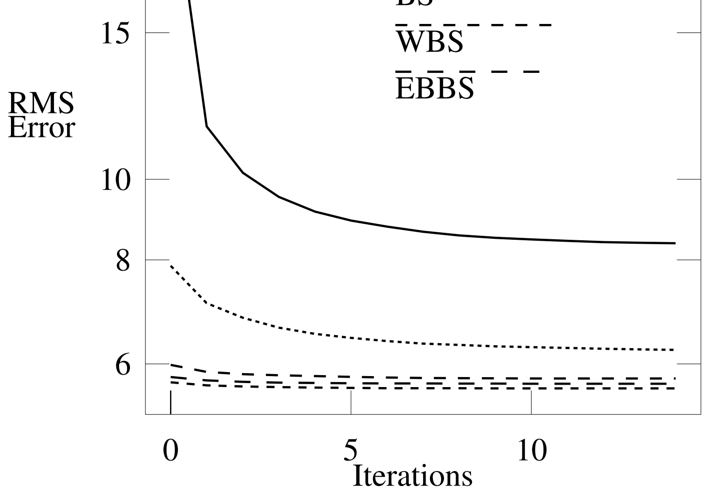

Figure 6: RMS Error versus the number of iterations of the LBG algorithm. Each plot corresponds to an initial color palette generated using one of the five quantization algorithms.

algorithms and then quickly reach a local minimum in RMSE. Although the LBG iterations offer significant improvements to the subjective quality of these algorithms, the displayed images produced by the resulting local minima still exhibit very objectionable false contouring in the case of the fixed (8,8,4) map and noticeable false contouring in the case of the palette generated by the HIST algorithm. The last three algorithms start near local minima and realize small improvements from the LBG iterations. For these algorithms, the LBG iterations produce no noticeable change in the subjective quality of the displayed images. Tests on all four images show that, although the LBG algorithm does reposition palette colors at each iteration, it reaches a local minimum before it can perform any major reallocation of colors in the color space. Because the general allocation of palette colors to regions in the color space remains the same before and after the LBG iterations, the quality of displayed images produced by the LBG algorithm is very dependent on the initial palette selected.

Table 2 compares the computational complexity of the algorithms by showing their execution times on a Sun Sparcstation and a rough measure of required multiplications. The execution times are averaged over the four images which were quantized. Since the three tree-structured algorithms perform pixel mapping during the palette design process, the times shown for the first two algorithms also include both palette design time and pixel mapping time. The time listed for the LBG algorithm reflects the execution time of a single iteration utilizing a recently proposed fast method for computing nearest neighbors [17]. The second column of Table 2 lists the number of multiplications used by each algorithm in terms of the number of pixels, N, and the number of palette colors, M. This computation count relates the complexity of the last three algorithms and the LBG algorithm by comparing the multiplications required for computing distances and autocorrelation matrix elements. However, this count does not measure the complexity of the first two algorithms since their computational complexity is dominated by operations other than multiplication. Since the multiplicative complexity of the fast algorithm in [17] varies significantly as a function of the map, the multiplication counts given for the HIST and LBG algorithms reflect the complexity of the full search algorithm for computing nearest neighbors instead of the algorithm used in the timing comparisons.

Algorithm	Execution Time (sec.)	Mult. Count
(8,8,4)	4.0	–
HIST	305.5	$3NM$
BS	56.3	$6N(\log_2 M + 1)$
WBS	127.6	$7.5N(\log_2 M + 1) + N$
EBBS	59.6	$6N(\log_2 M + 1)$
1 iteration of LBG	32.1	$3NM$

Table 2: Computational complexity of color quantization algorithms. Column 1 lists run times for each algorithm on a Sun Sparcstation, and column 2 lists the required number of multiplies for a image with N pixels and a palette with M colors.

The error diffusion methods for pixel mapping attempt to shape the error spectrum in order to minimize low-frequency error at the expense of higher RMS error. Figure 7 shows error spectra of quantized versions of Image 1 with and without error diffusion. Both plots use a 64 color palette generated by the EBBS algorithm. The algorithm of Figure 7a maps each color to the nearest neighboring palette color while the algorithm of Figure 7b uses the modified error diffusion method with ($\alpha = 6$) as described in Section 3.2. The plots show that, while nearest neighbor pixel mapping minimizes overall error power, error diffusion produces a shaped error spectrum with lower error energy at low frequencies. Table 3 shows this same result numerically by comparing these same two methods of pixel mapping using the the RMSE and a filtered RMSE (FRMSE) performance criteria. The FRMSE is computed by low pass filtering the error before computing the RMS deviation. This reflects the frequency sensitivity of the human visual system together with the spatial bandwidth of the display device.

Algorithm	RMSE	FRMSE
Nearest Neighbor	8.87	5.49
Error Diffusion	10.53	4.70

Table 3: RMSE and filtered RMSE (FRMSE) produced by the nearest neighbor and error diffusion methods of pixel mapping.

Section 3.2 discussed how the nonuniform distribution of colors in the palette can lead to quantization error oscillations and proposed the modified error diffusion method to eliminate these oscillations. One method of detecting oscillations is to measure the output error of the quantizer in the error diffusion loop. When this error is large it suggests that the input values to the quantizer are far from any available palette colors. In practice large errors result in very poor quality images often containing high frequency patterns of spots. Table 4 uses the image Peppers (the worst case of the four images) to show the dependence of the quantizer output RMSE on the value of the clipping parameter α and the method of quantization. It is useful to note the conventional error diffusion may be thought of as modified error diffusion with $\alpha = \infty$. The two types of quantization methods used are tree-based quantization, and a full search quantizer which finds the minimum distance palette color. In addition to the quantization RMSE, the percentage of pixels at which clipping occurs is also listed. This, together with the FRMSE, gives an indication as to when an excessive amount of clipping is occurring due to a small value of α. It is interesting to note that oscillation occurred both with and without the full search quantizer, and that when clipping was used the two

Min distance Power Spectral Density

(a)

MED Power Spectral Density

(b)

Figure 7: Spatial spectra of the quantization error in two displayed images. Each plot shows the RMS spectrum of the luminance component of the error. The plots correspond to the error spectrum from a) minimum distance pixel mapping and b) the modified error diffusion.

quantization methods had comparable performance. This table also indicates that $\alpha = 6$ yields a reasonable compromise between oscillation damping and restricted limiting.

Algorithm	Quantizer RMSE	% of pixels clipped	FRMSE
ED (tree search; $\alpha = \infty$)	426.9	0	8.24
ED (full search; $\alpha = \infty$)	242.4	0	6.68
MED (tree search; $\alpha = 6$)	26.4	8.1	4.89
MED (full search; $\alpha = 6$)	25.6	7.7	4.70
MED (tree search; $\alpha = 3$)	16.5	22.2	4.99
MED (full search; $\alpha = 3$)	16.0	21.7	4.83

Table 4: Performance comparisons of error diffusion and modified error diffusion methods for pixel mapping. The parameter α controls the amount of quantization error which can occur before the feedback signal is clipped.

In order to illustrate the artifacts present in the quantized images, we have included monochrome versions of some of the quantized images. These images were generated by using the luminance component of the quantized colors as defined in (6). While distortions in color are not visible, false contouring can still be seen. Each figure shows the center region of the image Lenna enlarged by a factor of two. This emphasizes smooth regions in her shoulder and face which are difficult to quantize. Figure 8 contains the results of the five color quantization algorithms, fixed (8,8,4), HIST, BS, WBS and EBBS. The fixed color map shows very severe degradation and contouring. The result of the histogram-based method is significantly better than the fixed map, but it still contains severe artifacts in areas of smooth color variation. The third image, generated using the basic binary splitting algorithm, has relatively good but not transparent quality. The areas above "Lenna's" lips and on her forehead and along her shoulder show false contours. The contours along her shoulders and face are significant when viewed on a high quality display. Both the WBS and EBBS algorithms are significant improvements over the basic algorithm, but the EBBS exhibited slightly more reliable performance with substantially less computation than the WBS algorithm. For 256 color palettes, we have found that both methods yield transparent quality for a variety of images. The advantage of the erosion method is that the actual quantization quality is measured during the palette design. In contrast, the weighting method predicts problematic areas of the image.

The results of two error diffusion algorithms using a 64 color EBBS palette are also shown in Figure 9. Both images used tree-based quantizers, but the first image used conventional error diffusion and the second used the modified error diffusion of Section 3.2 with $\alpha = 6$. The conventional error diffusion method resulted in saturated regions and spotty patterns caused by the large oscillations of the error diffusion filter. The modified error diffusion eliminated these artifacts and improved the display quality relative to minimum TSE pixel mapping. However, the image quality is still some what degraded relative to the original. We have found that error diffusion can dramatically improve the quality of an image generated from a excessively small palette, but it generally does not produce transparent results since in the process it adds some distinguishable amount of noise.

5 Conclusion

A variety of methods where explored for designing and using hierarchical image specific color palettes. The advantage of the hierarchical design is that nearly optimal color palettes can be designed in $\mathcal{O}(N \log M)$ time where N is the number of image pixels and M is the number of colors in the palette. In addition, the hierarchical structure allows colors outside the image to be quantized in $\mathcal{O}(N \log M)$ time. This computational savings is important in dithering applications. The color palette was improved by modifying the minimum total square error criteria to incorporate more subjective measures of image quality. This was done using two methods: subjective weighting and erosion. Erosion was found to require less computation, but both methods were found to yield very high quality results for palettes of size 256.

The techniques of ordered dithering and error diffusion were extended for use with image specific color palettes. Basic error diffusion exhibited instability due to the sparse nature of the color palette, but a modified error diffusion technique was proposed to limit this instability when it occurred. Modified error diffusion substantially reduced artifacts caused by excessively small palettes of size ranging from 32 to 128 colors.

References

[1] G. Braudaway, "A Procedure for Optimum Choice of a Small Number of Colors from a Large Color Palette for Color Imaging," *Electronic Imaging '87*, San Francisco, CA, 1987.

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

[3] Y. Linde, A. Buzo and R. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Commun.,* vol. COM-28, no. 1, pp. 84-95, Jan. 1980.

[4] R. Gentile, J. Allebach and E. Walowit, "Quantization of Color Images Based on Uniform Color Spaces," *Journal of Imaging Technology*, vol. 16, no. 1, pp. 11-21, Feb. 1990.

[5] R. Gentile, E. Walowit and J. Allebach, "Quantization and Multilevel Halftoning of Color Images for Near-Original Image Quality," *J. Opt. Soc. Am. A*, vol. 7, no. 6, pp. 1019-1026, June 1990.

[6] R. Gentile, E. Walowit and J. Allebach, "A Comparison of Techniques for Color Gamut Mismatch Compensation," to appear in *J. Imaging Technol.*

[7] C. Bouman, and M. Orchard, "Color Image Display with a Limited Palette Size," *Proc. of SPIE Conf. on Visual Comm. and Image Proc.*, pp. 522-533, Philadelphia, PA, Nov. 8-10, 1989.

[8] C. Bouman, *Hierarchical Modeling and Processing of Images.* Ph.D. thesis, Princeton University, October 1989, Department of Electrical Engineering.

[9] R. Gray, "Vector Quantization," *IEEE ASSP Mag.*, vol. 1, pp. 4-29, April 1984.

[10] R. Duda and P. Hart, *Pattern Classification and Scene Analysis,* Wiley, New York, NY, 1973.

[11] R. Floyd and L. Steinberg, "An Adaptive Algorithm for Spatial Gray Scale," *Proc. of the SID*, vol. 17, no. 2, pp. 75-77, 1976.

[12] A. Robertson, "The CIE 1976 Color-Difference Formulae," *Color Research and Application,* vol. 2, no. 1, pp. 7-11, Spring 1977.

[13] A. Netravali and B. Haskell, *Digital Pictures.* Plenum, New York, 1988.

[14] B. Rogowitz, "The Human Visual System: A Guide for the Display Technologist," *Proc. of the SID,* vol. 24, no. 3, pp. 235-252, 1983.

[15] J. Serra, *Image Analysis and Mathematical Morphology.* Academic Press Inc., Orlando, 1982.

[16] C. Billotet-Hoffmann and O. Bryngdahl, "On the Error Diffusion Technique for Electronic Halftoning," *Proc. of the SID,* vol. 24, no. 3, pp. 253-258, 1983.

[17] M. Orchard, "A Fast Nearest Neighbor Searching Algorithm with Applications in Vector Quantization," in preparation.

List of Figures

List of Tables

Original Fixedmap

HIST BS

a	b
c	d

Figure 8: A comparison of color palettes using enlarged monochrome versions of color images. Each palette contains 256 colors. a) Original image is shown together with quantized versions using b) an image independent color map of 8 red, 8 green and 4 blue; c) histogram based method (HIST); d) binary splitting method (BS); (continued on next page)

WBS; EBBS

e	f

Figure 8: e) weighted binary splitting (WBS); f) erosion based binary splitting (EBBS).

ED; MED

a	b

Figure 9: Monochrome versions of quantized color images using error diffusion methods. Each image uses a 64 color palette generated with the EBBS method. a) Result of conventional error diffusion, and b) modified error diffusion using $\alpha = 6$ and a tree search quantizer.