# USING COLOR VARIATION TO MIMIC HUMAN VISUAL PERCEPTION FOR COLOR IMAGE QUANTIZATION

**Yuchou CHANG**, **Dah-Jye LEE**, **James K. ARCHIBALD**

*Department of Electrical and Computer Engineering,*
*Brigham Young University, Provo, Utah, USA*
E-mail: ycchang@et.byu.edu, djlee@ee.byu.edu, jka@ee.byu.edu

### Abstract

Color quantization is an important part of image processing, image formation, and related research. This paper presents a novel color image quantization algorithm which addresses the question of how to incorporate principles of human perception of color variation into the process of quantization. The color variation measure, calculated first in CIE Lab color space, is used to evaluate color variation and to coarsely segment the image. Considering both color variation and homogeneity of the image, the number of colors that should be used for each segmented region can be determined. Finally, the CF-tree algorithm is applied to classify pixels into their corresponding palette colors. The quantized error of the proposed algorithm is small because it considers color variation and because it is based on a model of human visual perception. Experimental results suggest that the proposed algorithm is more effective in quantizing color images than a wide variety of previously proposed algorithms.

**Keywords**: color variation measure, color quantization, CIE color space, CF-Tree, human visual perception

## 1 Introduction

Color has long been recognized as critical in image and scene analysis. Research in color imaging has focused on a wide variety of topics, including color image formation, color quantization, human visual perception, image segmentation, color-based object recognition, and image database retrieval.

As a fundamental technique that is used widely in color image processing, color quantization plays an important role in many aspects mentioned above. In general, color image quantization can be divided into the following four phases [7]:

1. Sampling the original image for color statistics.
2. Choosing a color map based on color statistics.
3. Mapping original colors to their nearest neighbors in the color map.
4. Quantizing and representing the original image.

Because of the dependencies inherent between these four steps, the actions taken in Steps 1 and 2 can significantly impact the quality of the final results. This underscores the importance of selecting entries for the color map or palette that will be used to represent the entire image.

A good color quantization technique should consider several factors, including the complexity of the algorithm, the distortion in the resulting images, and the characteristics of the human visual system (HVS). To the best of our knowledge, no quantization approach described in the literature chooses the number of colors in the palette on the basis of a human visual perception model, or color variation in different image regions. In many techniques currently in use, a single palette is used for the whole image. However, a single image-wide palette cannot discriminate between different attributes in different regions of the image, even if those regions have color variations which affect the sensitivity of human visual perception.

One of the important HVS properties is that different colors and variations in their spatial patterns can influence the sensitivity of humans to color [11] [15]. To illustrate this principle, Fig. 1 shows images composed of the same set of colors, but with variations in the spatial patterns of those colors. It can be observed that increasing variations in the color spatial patterns – from left to right in Fig. 1) – cause a corresponding decrease in perceived sensitivity to color in the HVS. This suggests that more colors should be used to discriminate between color nuances in a homogeneous region, and fewer colors in regions with high levels of diversity.

The goal of palette design is to select a small number of representative colors from a high-resolution color image to form a color set. Using this color set or palette, the high-resolution color image can be represented by replacing the original colors with a smaller set of color elements chosen from the palette. The literature includes a variety of color quantization algorithms that have been proposed, including a median-cut algorithm [7], a popularity algorithm [7], a

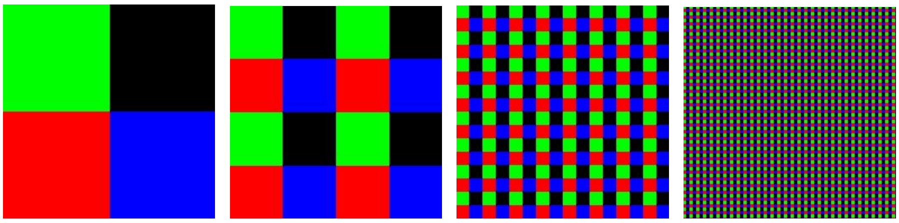


**Figure 1.** Test patterns of color visual perception sensitivity to spatial color pattern variation

variance-based algorithm [13], an octree algorithm [5], a center-cut algorithm [9], and a diversity algorithm [6]. Furthermore, a variety of clustering-based algorithms have been described, such as K-means, self-organizing maps [2], an adaptive clustering algorithm [8], and the peer group filtering method [3]. These algorithms are described in the following paragraphs.

In the median-cut algorithm [7],the colors of the original image are divided into $K$ boxes in the color space. Each box contains an equivalent number of pixels of the original image. The average color value of each box is then used as one of the $K$ entries in the color palette. Colors with many pixels can be included within one box, but colors with few occurrences cannot be represented as accurately. The popularity algorithm [7] generates a color map filled with the most common colors in the image. From a histogram of pixel values, the $K$ most frequently occurring colors are selected to fill the palette and to represent all colors in the image.

The variance-based algorithm [13] is similar to the median-cut algorithm. The difference lies in the iterative method of selecting each new entry for the color map; at each step, the cell or box with the largest weighted variance in terms of color distribution is chosen to be the next partitioned. The cutting plane for the partition is chosen to be perpendicular to the coordinate axis where the expected variance is most reduced. At the conclusion of this process, quantization errors in the resulting cells are nearly the same. The octree algorithm [5] subdivides the pixels from an image into a hierarchical structure of octants where placement is determined by RGB value. The closer two colors are to each other, the more likely they will be found in the same branch of the octree data structure. Once the octree is constructed, the $K$ leaf nodes serve as entries in the color map. The center-cut algorithm repeatedly splits the color set whose bounding box has the longest side until $K$ sets are generated. The centers of the $K$ sets are used as palette colors.

The diversity algorithm [6] quantizes the image using template color maps registered for sample document images. Images from batches to be classified are mapped onto the closest template color map using a fuzzy color classification algorithm. The self-organizing map (SOM) algorithm [2] uses a Kohonen neural network to adjust a quality factor to produce reduced colors of the image. The network maps an input $(R, G, B)$ to the index $i$, corresponding to a specific color map entry, which minimizes the error of quantization mapping. Clustering-based algorithms usually use minimal distance as the metric for clustering. The approach in [8] calculates a 3D-histogram which is fed into an adaptive clustering algorithm to extract palette colors from the image. A pixel-mapping algorithm then classifies each pixel by its corresponding palette color.

A few new quantization algorithms and extensions have been proposed in recent years. Zhao [16] proposed an improvement of the K-means algorithm. Atsalakis divided an image into small windows and quantized the major colors of these windows [1]. In [14], an algorithm integrated with Gamma correction was proposed and proved to be efficient to improve the visual effect of quantized images.

Color quantization is considered as a prerequisite for many color image segmentation algorithms [4]. Zouagui [17] proposed a new function model-based framework for image segmentation. This function model consists of five elementary blocks: measure, criterion, control, modification, and stop. Using the attributes from each segmented region, the segmented image is iteratively modified by the segmentation results from previous iterations.

In this paper, we propose a new color image quantization algorithm based on a measure of color variation. Based on principles presented in [11] and [15], the proposed algorithm considers the relationship between the segmented regions and their palette sizes so that regions which need more colors will be quantized into a greater number of color levels.

## 2 Color Variation Measure

### 2.1 Color Space Transformation

Color is perceived by humans as combinations of tristimuli $R$ (Red), $G$ (Green) and $B$ (Blue), or three primary colors. It is important to choose a good color space for the quantization of color images, since distance measures in color space are most effective when they conform to human visual perception. We note that the CIE color space was developed to represent perceptual uniformity. Furthermore, in CIE color space, color difference can be meaningfully calculated as the Euclidean distance between two color points.

In the CIE approach, colors are represented by combinations of three primaries: $X$, $Y$ and $Z$. Values of these primaries can be computed from RGB tristimulus coordinates using the linear transformation shown in Equation (1).

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.607 & 0.174 & 0.200 \\ 0.299 & 0.587 & 0.114 \\ 0.000 & 0.066 & 1.116 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{1}$$

The CIE color space, in turn represents colors using parameters $L$, $a$, and $b$, corresponding to the brightness of the color, its position between magenta and green, and its position between yellow and blue, respectively. Equations (2)-(4) show how the values of $L$, $a$, and $b$ can be computed from the values of $X$, $Y$, and $Z$ obtained using Equation (1). In the equations, $X_0$, $Y_0$, and $Z_0$ are the CIE XYZ tristimulus values of the reference white point.

$$L = 116\left(\sqrt[3]{\frac{Y}{Y_0}}\right) - 16 \tag{2}$$

$$a = 500\left(\sqrt[3]{\frac{X}{X_0}} - \sqrt[3]{\frac{Y}{Y_0}}\right) \tag{3}$$

$$b = 200\left(\sqrt[3]{\frac{Y}{Y_0}} - \sqrt[3]{\frac{Z}{Z_0}}\right) \tag{4}$$

## 2.2 Color Variation Measure Image

Color Variation Measure (CVM) is computed in both horizontal and vertical directions. In [12], directional operators were used to detect the edges of color images. Our approach adopts the directional operators to describe the color content variation within an image rather than to detect edges. The CVM for each pixel is composed of two positive and two negative components in each direction. The complete operation is described below, assuming a window sise of (2$s$+1)×(2$s$+1).

The RGB image is first transformed into CIE Lab color space according to Equations (1)-(4). For an image, the magnitude is normalized as:

$$\Delta\bar{H} = \begin{bmatrix} \bar{H}_1^- & & \bar{H}_1^+ \\ & \bar{0} & \\ \bar{H}_2^- & & \bar{H}_2^+ \end{bmatrix}, \quad \Delta\bar{V} = \begin{bmatrix} \bar{V}_1^- & & \bar{V}_2^- \\ & \bar{0} & \\ \bar{V}_1^+ & & \bar{V}_2^+ \end{bmatrix}.$$

These positive and negative components are defined, in turn, as follows:

$$\bar{H}_1^- = \frac{1}{s(s+1)} \sum_{y=y_0-s}^{y_0+1} \sum_{x=x_0-s}^{x_0+1} \bar{c}(x,y)$$

$$\bar{H}_1^+ = \frac{1}{s(s+1)} \sum_{y=y_0-s}^{y_0+1} \sum_{x=x_0-1}^{x_0+s} \bar{c}(x,y)$$

$$\bar{H}_2^- = \frac{1}{s(s+1)} \sum_{y=y_0-1}^{y_0+s} \sum_{x=x_0-s}^{x_0+1} \bar{c}(x,y)$$

$$\bar{H}_2^+ = \frac{1}{s(s+1)} \sum_{y=y_0-1}^{y_0+s} \sum_{x=x_0-1}^{x_0+s} \bar{c}(x,y)$$

$\bar{V}_1^-$, $\bar{V}_1^+$, $\bar{V}_2^-$, and $\bar{V}_2^+$ can be calculated in a manner similar to $\bar{H}_1^-$, $\bar{H}_1^+$, $\bar{H}_2^-$, and $\bar{H}_2^+$, respectively. In the formulas, $\bar{c}(x,y)$ denotes the color value $(L,a,b)$ at pixel $(x,y)$.

In order to get the color variation in the horizontal and vertical directions, we calculate the four vectors below.

$$\begin{aligned}
\Delta\bar{H}_1(x_o,y_0) &= \bar{H}_1^+(x_0,y_0) - \bar{H}_1^-(x_0,y_0)\\
\Delta\bar{H}_2(x_o,y_0) &= \bar{H}_2^+(x_0,y_0) - \bar{H}_2^-(x_0,y_0)\\
\Delta\bar{V}_1(x_o,y_0) &= \bar{V}_1^+(x_0,y_0) - \bar{V}_1^-(x_0,y_0)\\
\Delta\bar{V}_2(x_o,y_0) &= \bar{V}_2^+(x_0,y_0) - \bar{V}_2^-(x_0,y_0)
\end{aligned}$$

The scalars $\left\|\Delta\bar{H}_1(x_o,y_o)\right\|$, $\left\|\Delta\bar{H}_2(x_o,y_o)\right\|$, $\left\|\Delta\bar{V}_1(x_o,y_o)\right\|$, and $\left\|\Delta\bar{V}_2(x_o,y_o)\right\|$ are measures of color variation in the horizontal and vertical directions. In terms of these scalars, we define the color variation at pixel $(x_0,y_0)$ as follows:

$$M(x_0,y_0) = \sqrt{\left\|\Delta\bar{H}_1(x_0,y_0)\right\|^2 + \left\|\Delta\bar{H}_2(x_0,y_0)\right\|^2 + \left\|\Delta\bar{V}_1(x_0,y_0)\right\|^2 + \left\|\Delta\bar{V}_2(x_0,y_0)\right\|^2} \quad (5)$$

We use $M(x_o,y_0)$ as the color variation measure (CVM) to depict the variations in interior colors of an image. Fig. 2 shows the CVM values of the flower garden image represented in 256-level gray scale. Fig. 2(b) and 2(c) are the CVM images using a 3×3 and 5×5 window, respectively. In the images, areas with the greatest variation appear with the lowest grayscale intensity.

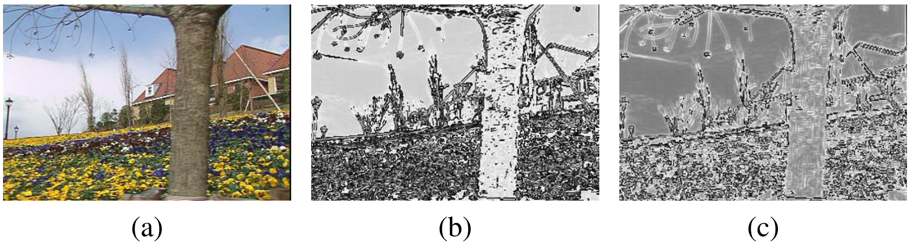


**Figure 2.** CVM image of the “Flower Garden” image. (a) Original image. (b) CVM image using a 3×3 window. (c) CVM image using a 5×5 window.

# 3 Algorithm

## 3.1 Framework

We propose a new color image quantization algorithm based on the Color Variation Measure. Fig. 3 shows a flow chart describing essential stepts in the algorithm. The CVM image block is the core of the algorithm. Not only is CVM a prerequisite of segmentation, but it also guides the quantization of different regions based on color variation. Details of the algorithm are discussed in the following sections.

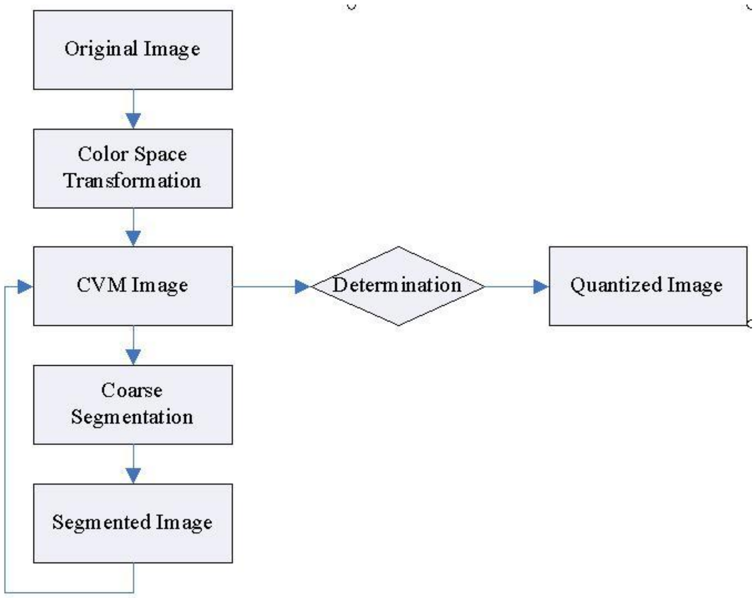


**Figure 3.** Color image quantization algorithm.

## 3.2 Coarse Segmentation

One goal of our algorithm is to use principles of human visual perception to color variation to guide the quantization of color images. To achieve this goal, we adopt a region-growing and merging-based algorithm to coarsely segment the image. For region growing and merging, it is necessary to specify seeds as the starting points of regions, and the seed choices directly influence the final outcome of segmentation. If the chosen seeds do not typify the attributes of the surrounding regions, the regions produced by segmentation will not be correct and the segmentation process will be prolonged.

In our approach, we use the CVM image to quantize values of all pixels to gray levels between 0 and 255. According to the quantized CVM values, we can use the method in [10] to obtain a threshold $T$, or it can be specified by the user. If the CVM value of a pixel is less than $T$, it is specified to be a seed point, which can be merged with neighboring pixels to form a candidate seed area. If the number of pixels in candidate seed area exceeds the corresponding number in Table 1, it will be used as a seed area in the region growing algorithm.

**Table 1.** Minimum pixels to form seed area for image size

| Image Size (min(width,height)) | Min Seed (pixels) |
|---|---|
| $\leq 128$ | 64 |
| $\leq 256$ | 256 |
| $\leq 512$ | 1024 |

After the identification of seed areas, all pixels are grouped into two categories: seed area pixels and non-seed area pixels. The non-seed area pixels must be merged into their corresponding segmented regions. The strategy to achieve this is as follows:

1. If a non-seed area is adjacent to only one seed area, it is merged into that seed area. This removes non-seed areas that are surrounded by a single seed area.

2. If a non-seed area is adjacent to more than one seed area, it will be merged into the seed area that has the shortest distance in CIE Lab color space.

3. In each iteration, the mean color of every seed area is recomputed until there is no non-seed area.

Areas with fewer than 50 pixels will be merged into adjacent areas that are larger in size. The segmentation results of the CVM images (Fig. 2(b) and 2(c)) are shown in Figs. 4(a) and 4(b), respectively.

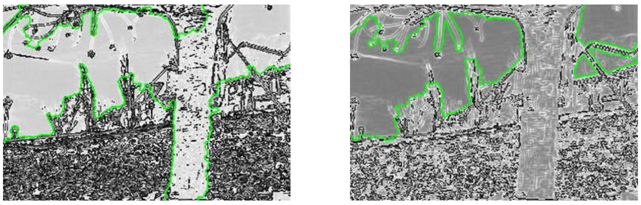

(a) (b)

**Figure 4.** Segmentation results for the “Flower Garden” image. (a) CVM image using a 3x3 window. (b) CVM image using a 5x5 window.

## 3.3 CVM-based Color Quantization

After coarse segmentation, each pixel can be assigned a class label, forming a class map. In the class map, pixels are grouped into different regions with corresponding labels. Because each segmented region is not necessarily homogenous, the degree of internal color variation can differ widely between regions. Therefore, calculating the statistics of pixel CVM distributions of different regions can assist in judging whether the number of quantization levels is valid. If one segmented region has little color variation, it should be assigned more quantization levels. Because the sensitivity of human visual perception increases as color variation decreases, the use of more color quantization levels will result in more accurate color representation to a human observer. For regions with high levels of color variation, fewer color levels should be assigned.

In each segmented region, CVM values of all pixels are used to calculate the region’s homogeneity. Assuming that the image has been segmented into $m$ regions, we define $\lambda_k$, the degree of homogeneity for region $k$, denoted $R_k$, as follows:

$$\lambda_k = \sum_{p(i,j)\in R_k} \varepsilon_{p(i,j)}$$

where $p(i, j)$ is the pixel at image coordinates $(i, j)$ and $\varepsilon$ is given by

$$\varepsilon_{p(i,j)} = \sum_{\substack{p(x,y)\in R_m \\ x=i\pm1, y=j\pm1}} \left| M_{p(i,j)} - M_{p(x,y)} \right| .$$

In the equation, $M$ refers to the CVM measure defined by Equation (5).

In general, the smaller the value of $\lambda$, the more homogenous the region, and the more colors should be assigned. In order to maintain the integrity of the color perception of the whole image, we need to consider both the degree of color variation degree and the degree of homogeneity simultaneously. First, we use the K-means algorithm on the whole image to get an initial number of colors for every segmented region $N_{K1}, N_{K2}, \ldots, N_{Km}$. Second, we use CVM to adjust the number of colors for each segmented region.

$$N_{Ci} = (int)\left(1 - \frac{\lambda_i}{\sum_{j=1}^{m} \lambda_j}\right) N$$

To obtain the final number of colors for each region of the resulting image while maintaining the integrity of color perception of the whole image, three cases must be considered.

1. If $N_{Ki} > N_{Ci}$, color variation and complexity in region $R_i$ is above average for the image, and fewer colors should be used. The assigned number of colors is given by:

$$N_i = N_{Ci} + \frac{N_{Ki} - N_{Ci}}{2}, \quad i = 1, 2, \ldots, m$$

2. If $N_{Ki} < N_{Ci}$, color variation and complexity in region $R_i$ is below average for the image, and more colors should be used. The number of colors to be used is given by:

$$N_i = N_{Ki} + \frac{N_{Ci} - N_{Ki}}{2}, \quad i = 1, 2, \ldots, m$$

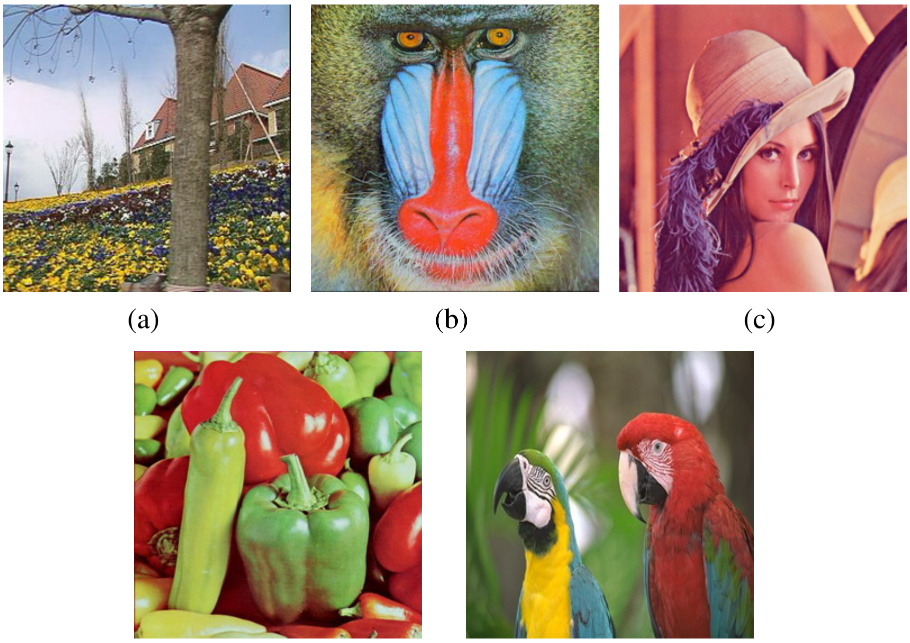
3. If $N_{Ki} = N_{Ci}$, the number of colors need not be increased or decreased:

$$N_i = N_{Ki} = N_{Ci}$$

Once it has been determined how many colors are to be used in each region, we use the CF-tree method [14] to quantize each segmented region according to its number of colors.

## 4 Experimental Results

To evaluate the performance of the proposed CVM-based color quantization algorithm, we use five 24-bit 256×256 images for experiments. These images are: “Flower Garden”, “Baboon”, “Lena”, “Peppers”, and “Parrots”, as shown in Fig. 5.


(a) (b) (c)

(d) (e)

**Figure 5.** Images for experiments. (a) Flower Garden; (b) Baboon; (c) Lena; (d) Peppers; and (e) Parrots.

To compare the difference between the original image and the quantized image, we use mean squared error (MSE), defined as follows:

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} d(c[i, j], q(d[i, j]))^2 \qquad (6)$$

In Equation (6), $c[i, j]$ are original pixels, $q(c[i, j])$ are quantized pixels, $MN$ is the total number of pixels in the image, and $d(x, y)$ is the Euclidean distance between color $x$ and color $y$. Smaller values of MSE mean better image representation after color quantization. The five tested images were quantized into 128, 64, 32 and 16 colors for comparison. Algorithms implemented for comparison included median-cut [7], popularity [7], K-means, variance-based [13], diversity [6], SOM [2], and octree [8]. As representative examples, Figs. 6, 7, and 8 show the quantization results of the "Flower Garden" using 128 colors, "Baboon" using 64 colors, and "Parrots" using 32 colors, respectively.

As can be seen in Fig. 6, the CVM-based method quantizes the homogeneous blue sky region using more colors than the flower and garden regions that have greater variations in color. From the images in Figs. 7 and 8, it can be noted that the proposed CVM-based method can discriminate detailed color information in homogeneous regions to achieve good quantization results that are close to human color perception.

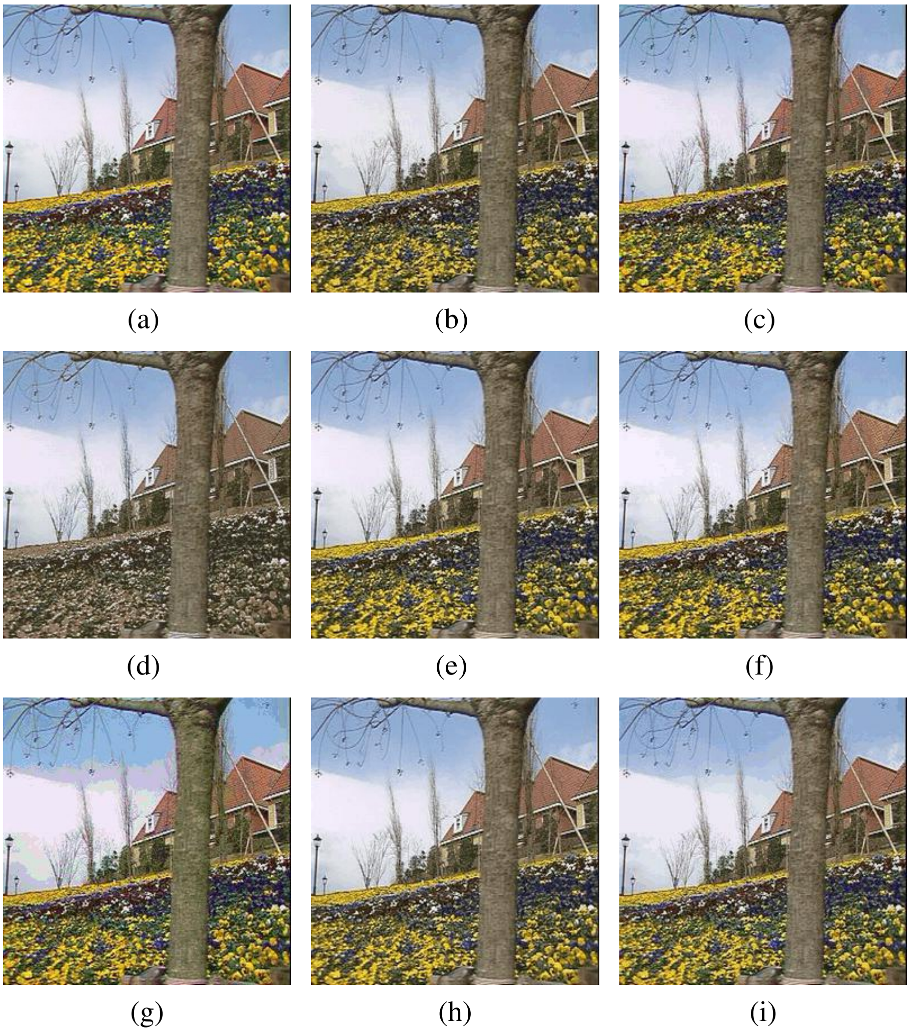

(a) (b) (c) (d) (e) (f) (g) (h) (i)

**Figure 6.** "Flower Garden" images of 128 colors: (a) Original image; (b) The proposed CVM-based method; (c) Median cut; (d) Popularity; (e) K-means; (f) Variance-based method; (g) Diversity; (h) SOM; (i) Octree.

Table 2 shows the performance measured by MSE over all tested images. MSE values of all five test images of 128, 64, 32 and 16 colors are listed. The last row of Table 2 shows the average percentage advantage of the proposed CVW-base method measured by MSE. The proposed CVM-based algorithm is

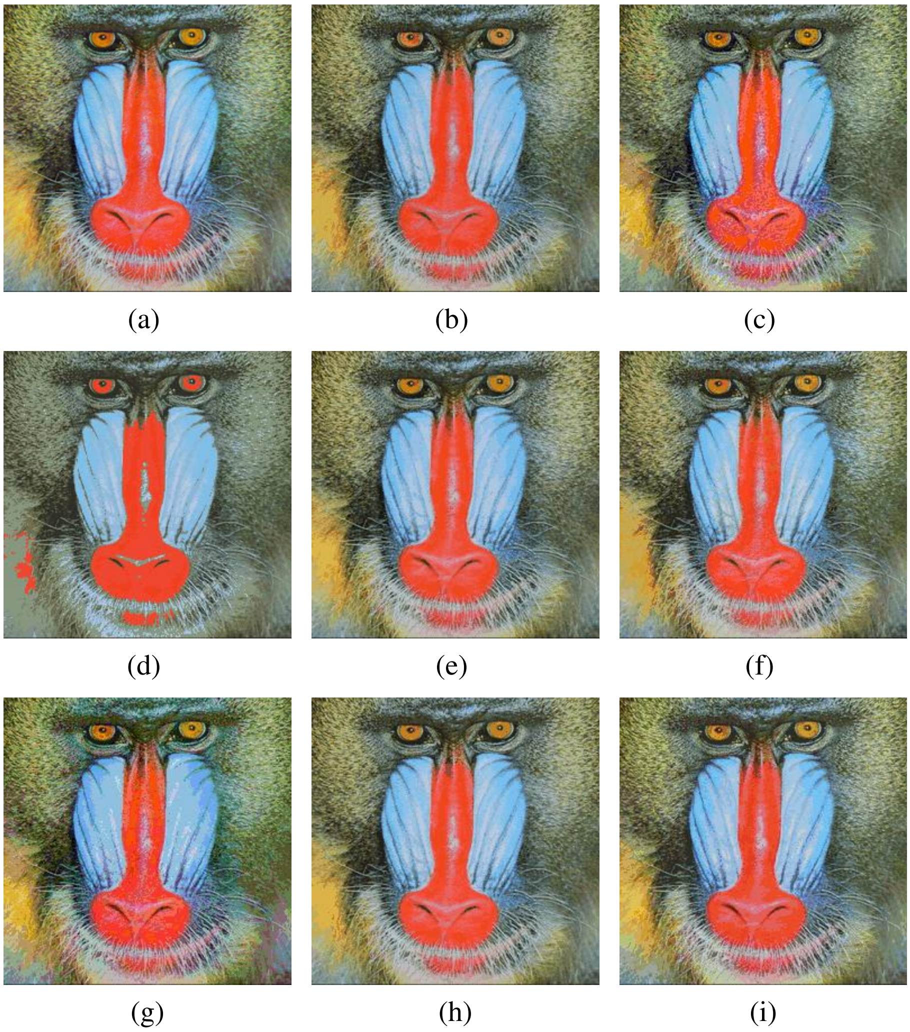
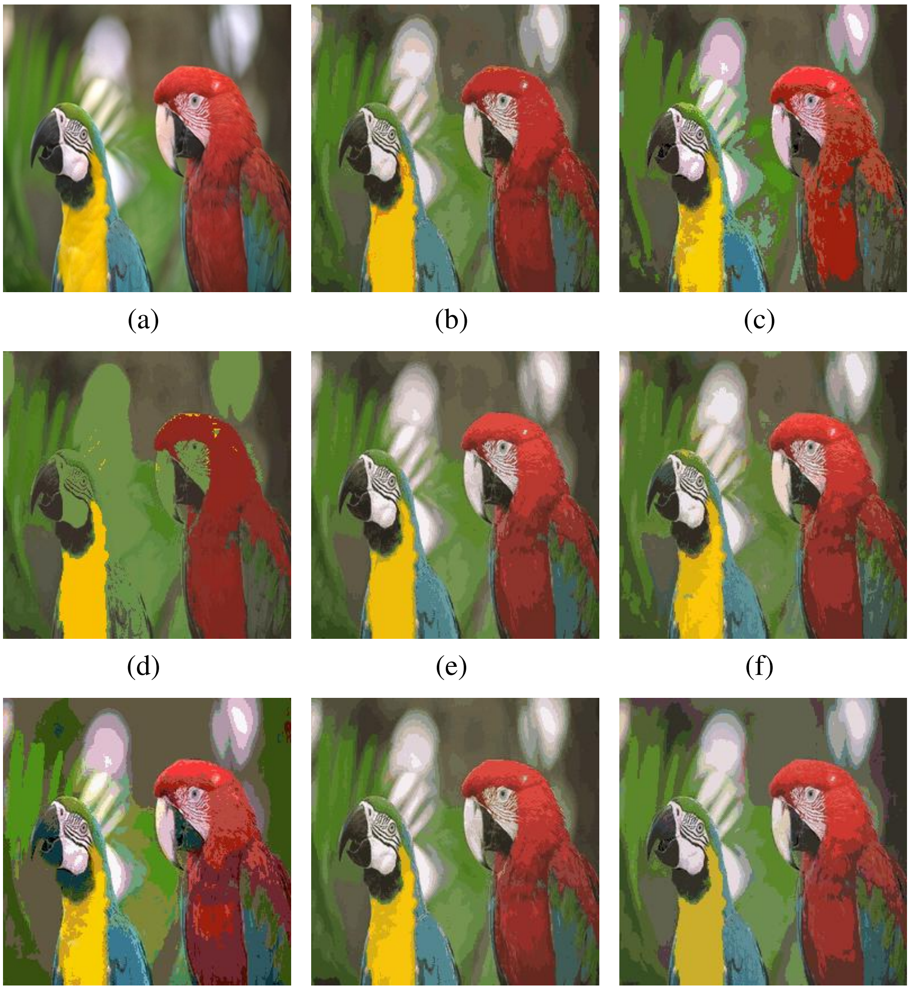


**Figure 7.** “Baboon”’ images of 64 colors: (a) Original image; (b) The proposed CVM-based method; (c) Median cut; (d) Popularity; (e) K-means; (f) Variance-based method; (g) Diversity; (h) SOM; (i) Octree.

slightly better than Median-cut (2.9%) and K-means (4.2%) and significantly better than the rest of the methods, ranging from 13.6% better than SOM to 47.7% better than the diversity method.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

**Figure 8.** “Parrots” images of 32 colors: (a) Original image; (b) The proposed CVM-based method; (c) Median cut; (d) Popularity; (e) K-means; (f) Variance-based method; (g) Diversity; (h) SOM; (i) Octree.

## 5 Conclusion

We propose a quantization algorithm for color images which considers the sensitivity of human visual perception to color variation. The algorithm first transforms the image from RGB color space into CIE Lab color space and then calculates horizontal and vertical color variation measure (CVM) values. Secondly, region growing and merging algorithms are used to coarsely seg-

**Table 2.** Comparative performance in terms of MSE values on 24-bit color images

| Image | No. Colors | CVM-based | Median cut | K-means | SOM | Popularity | Var.-based | Octree | Diversity |
|---|---|---|---|---|---|---|---|---|---|
| Flower Garden | 16 | 93.9 | 96.6 | 100.8 | 104.3 | 125.0 | 111.3 | 133.5 | 157.0 |
| | 32 | 76.1 | 82.3 | 77.2 | 85.0 | 101.1 | 96.5 | 106.0 | 140.0 |
| | 64 | 57.3 | 63.7 | 56.6 | 62.5 | 80.9 | 75.2 | 95.5 | 108.4 |
| | 128 | 38.4 | 39.5 | 40.7 | 42.3 | 63.7 | 57.1 | 62.2 | 76.6 |
| Baboon | 16 | 119.8 | 122.9 | 125.0 | 130.2 | 136.5 | 129.9 | 129.0 | 148.2 |
| | 32 | 81.0 | 76.1 | 77.2 | 85.0 | 101.1 | 96.5 | 106.0 | 140.0 |
| | 64 | 76.3 | 73.8 | 72.7 | 80.2 | 93.8 | 92.3 | 106.8 | 122.3 |
| | 128 | 46.6 | 48.2 | 51.3 | 57.1 | 68.8 | 74.0 | 79.0 | 84.4 |
| Lena | 16 | 52.9 | 67.6 | 72.2 | 79.8 | 49.8 | 87.1 | 130.4 | 125.5 |
| | 32 | 40.6 | 46.0 | 43.8 | 51.0 | 43.3 | 67.2 | 94.3 | 110.2 |
| | 64 | 31.7 | 26.4 | 28.3 | 56.1 | 35.1 | 46.9 | 56.1 | 87.1 |
| | 128 | 17.2 | 15.2 | 14.8 | 21.9 | 18.9 | 33.9 | 38.6 | 46.9 |
| Peppers | 16 | 98.6 | 103.9 | 106.8 | 106.1 | 138.0 | 111.2 | 131.0 | 144.3 |
| | 32 | 83.3 | 92.2 | 81.7 | 83.6 | 100.5 | 91.6 | 136.5 | 131.5 |
| | 64 | 55.0 | 57.9 | 58.8 | 60.9 | 63.9 | 71.9 | 101.1 | 110.6 |
| | 128 | 37.9 | 36.1 | 40.2 | 35.6 | 37.2 | 53.7 | 82.4 | 71.9 |
| Parrots | 16 | 90.3 | 87.2 | 100.3 | 101.9 | 93.6 | 118.5 | 118.1 | 159.5 |
| | 32 | 49.2 | 53.3 | 74.6 | 74.1 | 50.6 | 89.9 | 114.3 | 137.0 |
| | 64 | 46.2 | 48.2 | 50.3 | 54.8 | 49.9 | 66.6 | 85.3 | 105.3 |
| | 128 | 35.9 | 42.9 | 33.3 | 39.3 | 44.2 | 48.7 | 62.2 | 72.1 |
| Avg % | - | - | **2.9** | **4.2** | **13.6** | **15.5** | **26.6** | **39.2** | **47.7** |

ment the CVM image to obtain the class label for each pixel and determine the number of colors in each region. Third, a K-means approach is used to adjust the number of colors to be used in each region. Adjustments at this step maintain the integrity of the color balance of the whole image. Finally, the CF-tree method is adopted to quantize each region using the calculated number of colors.

In quantizing color images, a single image-wide palette can not always perform well for both homogeneous and inhomogeneous regions, because human visual perception of these regions is different. The proposed algorithm mimics human visual perception and adjusts the color palette in a region-specific manner based on local variations in color. Our experimental results demonstrate that the proposed CVM-based algorithm is superior to other previously proposed quantization algorithms. In future work, we will explore the use of this novel color quantization method for other image processing problems such as color image segmentation, content-based image retrieval, and image compression.

## References

[1] Atsalakis, A., Kroupis, N., Soudris, D., Papamarkos, N., 2002, *A window-based color quantization technique and its embedded implementation*, Proc. of Int. Conf. on Image Processing, Rochester, NY, pp. 365-368.

[2] Dekker, A.H., 1994, *Kohonen neural networks for optimal colour quantization*, Network: Computation in Neural Systems. Vol. 5, No. 3, pp. 351-367.

[3] Deng, Y., Kenney, C., Moore, M.S., Manjunath, B.S., 1999, *Peer group filtering and perceptual color image quantization*, Proc. IEEE Int. Symp. on Circuits and Systems, Vol. 4, Orlando, FL, pp. 21-24.

[4] Deng, Y., Manjunath, B.S., 2001, *Unsupervised segmentation of color-texture regions in image and video*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 23, No. 8, pp. 800-810.

[5] Gervautz, M., Purgathofer, W., 1990. *A simple method for color quantization: octree quantization*, In: Graphics Gems, Academic Press Professional, Inc., San Diego, CA, pp. 287-293.

[6] He, J., and Downton, A.C., 2004, *Color map classification for archive documents*, Lecture Notes in Computer Science, Vol. 3136, pp.241-251.

[7] Heckbert, P., 1982, *Color image quantization for frame buffer display*, Computer Graphics, Vol. 16, No. 3, pp. 297-307.

[8] Hsieh, I.S., Fan, K.C., 2000, *An adaptive clustering algorithm for color quantization*, Pattern Recognition Letters, Vol. 21, No. 4, pp. 337-346.

[9] Joy, G., Xiang, Z., 1993. *Center-cut for color image quantization, Visual Computation*, Vol. 10, No. 1, pp. 62-66.

[10] Pun, T., 1981, *Entropic thresholding: a new approach*, Computer Graphics and Image Processing, Vol. 16, No. 3, pp. 210-239.

[11] Schnapf, J.L., Kraft, T.W., Baylor, D.A., 1987, *Spectral sensitivity of human cone photoreceptors*, Nature, Vol. 325, pp. 439-441.

[12] Scharcanski, J., and Venetsanopoulos, A. N., 1997, *Edge detection of color images using directional operators*, IEEE Trans. on Circuits and Systems for Technology, Vol.7, No. 2, pp. 397-401.

[13] Wan, S.J., Wong, S.K.M., Prusinkiewicz, P., 1988, *An algorithm for multidimensional data clustering*, ACM Trans. Math. Software, Vol. 14, pp. 153-162.

[14] Zhang, T., Ramakrishnan, R., Livny, M., 1996, *BIRCH: An efficient data clustering method for very large databases*, Proc. of ACM SIGMOD Int. Conf. on Management of Data, Montreal, Quebec, pp. 103-114.

[15] Zhang, X., Silverstein, D.A., Farrell, J.E., Wandell, B.A., 1997, *Color image quality metric S-CIELAB and its application on halftone texture visibility*, Proc. of IEEE Compcon, San Jose, CA, pp. 44-48.

[16] Zhao, Y.W., Wang, W.L., 2000, *A new clustering algorithm for color quantization and its applications*, Journal of Computer Aided Design & Computer Graphics, Vol. 12, No. 5, pp. 340-343.

[17] Zouagui, T., Benoit-Cattin, H., Odet. C., 2004, *Image segmentation functional model*, Pattern Recognition, Vol. 37, No. 9, pp. 1785-1795.