Design and Performance Analysis of Hierarchical Color Space Quantizers

Xia Wan, Jiankun Li, Yung-Kai Lai and C.-C. Jay Kuo
Integrated Media Systems Center, University of Southern California
Los Angeles, California

Abstract

Color space quantization, which maps all points in a color space to a small set of representative points, is essential in many applications such as graphic rendering and image retrieval based on color features. In this research, we propose and compare several hierarchical quantization schemes in different color spaces. Examples from image retrieval and cartoon rendering are used to compare the performances of different proposed quantizers which include computational complexity, correct retrieval rate, and human visual effect.

I. Introduction

Digital images are often represented in the RGB space with 8 bits for each primary color of R,G, and B. Thus, color distribution of an image is defined on a cubic lattice consisting of 256×256×256 (16.8 millions) points. However, for applications such as color image retrieval and graphic rendering, such a fine resolution is often more than necessary. It is more economical to perform color space quantization, i.e. mapping these points to a much smaller set of representative points. In this research, we propose several ways to design hierarchical quantizers which partition a given color space into smaller subspaces with a coarse-to-fine approach. With such vector quantization (VQ) schemes, we are able to stop the quantization process at a certain desired level, and save unnecessary storage and computation. The most important feature of color quantizer design is to select the code vectors as the centroid of training vector clusters, where a good distance measure is required. In this paper, we investigate the distance measure in a space which takes into account the human perception response.

Most natural images consist of complicated textured patterns, numerous possible colors, and rich variations of intensity levels. Compared to natural images, cartoon images have several features, scene simplicity, less number of colors and color uniformity. They could be potentially exploited to achieve a better rendering performance. It is well known that the colors of a cartoon images are either uniform or changing regularly within a certain region. There are however some thin strokes scattering inside a region, which give the meaning of the scene, such as eyes, nose and mouth strokes in an otherwise blank face. It is very important to keep these details, which are essential for the recognition and entertainment purpose. So far, there is no rendering scheme developed to explore these features. In this work, we develop a simple but efficient coding scheme for the rendering of cartoon images by taking there special characteristics of cartoon images into account.

This paper is organized as follows. In Section II, hierarchical color quantization schemes are introduced. The human visual response to color differences is discussed in Section III. Applications in color image retrieval and rendering are presented in Section IV. Experimental results are given in Section V to demonstrate the performance of proposed hierarchical color quantization schemes. Concluding remarks are given in Section VI.

II. Color Quantization and Applications

Color Spaces

There are several commonly used color spaces such as CIE, RGB, HSV, CIELUV, etc. Our design and analysis are carried out in these typical color spaces based on statistical distributions in these spaces.[5] In other words, color distributions of the entire database on these color spaces are analyzed to develop a good quantization strategy.

(1) RGB Color Space

Digital images are normally represented in the RGB space. The distribution of each component R, G, B is flat except at the two end points which correspond to the background of images. This corresponds to a near-uniform color distribution in each of these axes so that we can use uniform quantization for each of the color axes. Because R, G and B have the same contribution to the color sensitivity of human eyes, the quantization steps of R,G,B should be same.

(2) HSV Color Space

HSV (Hue, Saturation, Value) is an intuitive color space in the sense that each component contributes directly to the visual perception. Other closely related color spaces include HSI (I: Intensity), HSL (L: Lightness) etc. Since the Human Visual System (HVS) is more sensitive to H than S and V, H should be quantized finer than S and V.

(3) YUV Color Space

In the YUV space, Y represents the luminance component of a color whereas U and V represent the chromaticity components of color. One advantage of the YUV space is that the luminance is separated from chrominance, which is a feature useful in compression and image processing. U and V are approximately Laplacian distributed and Y is nearly uniform. Thus the quantization should not be uniform in the U and V coordinates to reduce the quantization error.

(4) Munsell Color Space

There are a lot of effort (e.g., Munsell, CIELUV, CIELAB) to linearize the perceptibility of color differences. These color spaces have two features: scale linearity and psychological independence of one coordinate to the other.

Quantizer Design

Several commonly used color space quantizers are summarized below:

(1) Uniform Quantizer

Each axis of the color space is uniformly subdivided into pre-specified number of bins with each bin of the same size. The advantage of the uniform partitioning scheme is that it is straightforward and is the simple choice in the absence of information regarding the color distribution of the database images. It introduces a large quantization error when the color distribution is highly nonuniform.

(2) LBG Quantizer

The LBG or the k-means quantizer is a method to partition the vector space by the minimization of the mean-squared error (MSE). When applied to color space quantization, it treats each pixel of images in the database as a vector in the 3-D color space and the quantizer optimally subdivides the color space into a specified number of subspaces so that the resulting MSE is minimized. The biggest advantage of k-means quantization is the optimum partitioning of the color space so as to use the number of histogram bins effectively. However, the main disadvantage is the increased processing complexity.

(3) Product Vector Quantizer

Although k-means quantizers result in the optimum partitioning of color spaces, it is inefficient as a quantization scheme for image databases due to the computational complexity. A simpler quantization method is to divide the color space where the partition is perpendicular to each axis of the color space. This can be done by a method known as product VQ. In particular, a Lloyd-Max quantizer can be applied to each axis of the color space separately to optimize the mean squared quantization error along each axis independently. Thus, the quantization scheme subdivides each axis separately according to its individual color distribution. The advantage of such a color quantization scheme is that the computational complexity of the histogram extraction is greatly reduced. Thus, product VQ with the Lloyd-Max quantizer results in a better partitioning of the color space as compared to uniform quantization without a tremendous increase in computational complexity.

Hierarchical Quantizer

Hierarchical quantizers can be designed based on a generalization of these three approaches. For uniform and product VQ quantizers, we can adopt a split-and-merge process to partition the color space into an octave-tree structure. For LBG quantizers, the tree-structured vector quantization (TSVQ) technique can be exploited to reorganize the data points in the 3-D color space into a hierarchical tree. However, its computational complexity is usually much higher.

III. Human Vision Factors

To implement a good vector quantizer which is consistent with HVS model, we would like to find a metric which measures color differences (or color distance) with respect to human perception responses. Quantitative measurement of the color distance has been studied extensively, and psychophysical experiments has been conducted to determine the Just Noticeable Color Differences (JNCD) in an ordinary color space such as RGB or XYZ spaces.[1] However, as we would expect, JNCD is not uniform along the three axes in an ordinary color space. Therefore, the difference between two colors cannot be directly computed as the Euclidean distance between these two colors. The infinitesimal color difference ds of two neighboring colors can written as

$$ds^2 = \sum_{i,j=1}^{3} C_{i,j}\, dx_i dx_j \qquad (1)$$

where metric coefficients depend on x_i..

To find the difference between two colors, we have to integrate Equation (1) from one color to the other. The integral is path-dependent and the actual distance is the integral along the path which yields the minimum distance between the two colors. Methods such as dynamic programming and steepest descent methods[2] has been used to determine the geodesics.

Since the computation costs in 2 is quite high, an alternative is to map the traditional space to another space with a uniform color difference, and to find the transformation between the original and the mapped color spaces. Several such spaces has been proposed, CIELAB and CIELUV are two most prominent systems which both became CIE standards in 1976. In these spaces, color differences in an arbitrary direction are approximately equal. Thus the relative color distance of two colors can be determined by the Euclidean distance between them in this space. Some quantitative analysis[3] indicated that these two spaces have approximately the same uniformities of color differences. However, it is noted that for applications involving self-luminous displays, LUV has been the more popular one of the two since it is the more linear than LAB. Another reason that we should choose LUV over LAB is the convergence issue in VQ. In RGB space, the color gamut is the unit cube, in which the General Lloyd-Max Algorithm (GLA) of VQ is guaranteed to converge since the domain is convex. The same guarantee is also ensured in other spaces such as XYZ, YIQ and YUV, which are linear transformed from the RGB space so that their color gamuts are also convex. However, LAB and LUV spaces do not have a linear relations with the RGB space. From the color gamut shown in Figure 1, we see that the LAB gamut is highly concave, while the LUV gamut is near-convex. Therefore, it is quite unlikely for the resulting code vectors to fall outside the LUV gamut in practical situations. If this happens, we can still project the code vector back to the nearest point in the gamut, which should be close enough to the optimal solution.

IV. Applications

Color Image Retrieval

The color histogram of an image describes its color distribution, where every color in the image corresponds to a point in a 3-D color space. Each pixel in a 24-bit image is one of possible colors. As explained earlier, a such fine color resolution is impractical for image retrieval. Therefore, the color space can be quantized to reduce the color resolution or the color depth. The quantized color histogram is then used along with suitable color similarity measures to aid image retrieval.

Global color histogram is widely used to facilitate retrieval process because of its robustness with respect to scaling, orientation, perspective and occlusion.[6] It provides a good approach for retrieving images that have similar overall color content. The archived images can be sunsets, beach scenes, ski scenes etc. There are two aspects in color-distribution-based retrieval: the derivation of histograms, and the distance metric. Color quantization scheme is critical to image retrieval based on color features. The effects of different quantization schemes on the performance of retrieval are investigated in this work. The histogram intersection method[7] is used to measure the similarity of histograms. Experimental results are shown in Section 5.

Cartoon rendering

The performance of image rendering is affected by many factors. One of them is that the monitor cannot provide all the necessary colors to fully render the image. To reduce the number of required colors, neighboring colors have to be combined and represented with a single color so that the whole image is rendered by a smaller number of colors. This procedure is equivalent to the VQ process in the color space. Every color in the image can be taken as a sample point in the color space. Then, we do the quantization in the color space. Each centroid represent a color which can be used to replaced any color samples inside this cell. We perform experiment with many combinations of VQ schemes and color space representation discussed earlier, and then use both their objective MSE's and subjective visual performances to evaluate these quantizers. The MSE measures the overall fidelity of VQ-colored image from full-colored image while the visual investigation emphasizing the resemblance of some specific colors and artistic details. Experimental results are shown in Section 5.

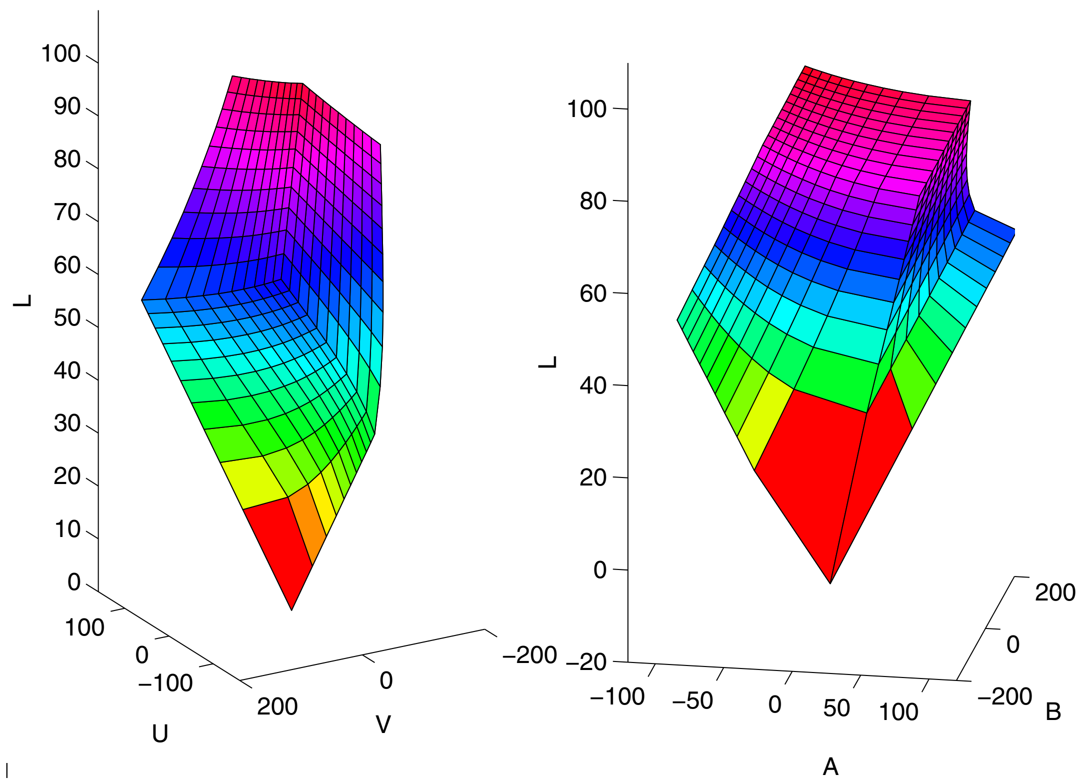

Figure 1. The gamuts of LUV (Left) and LAB (Right) color spaces

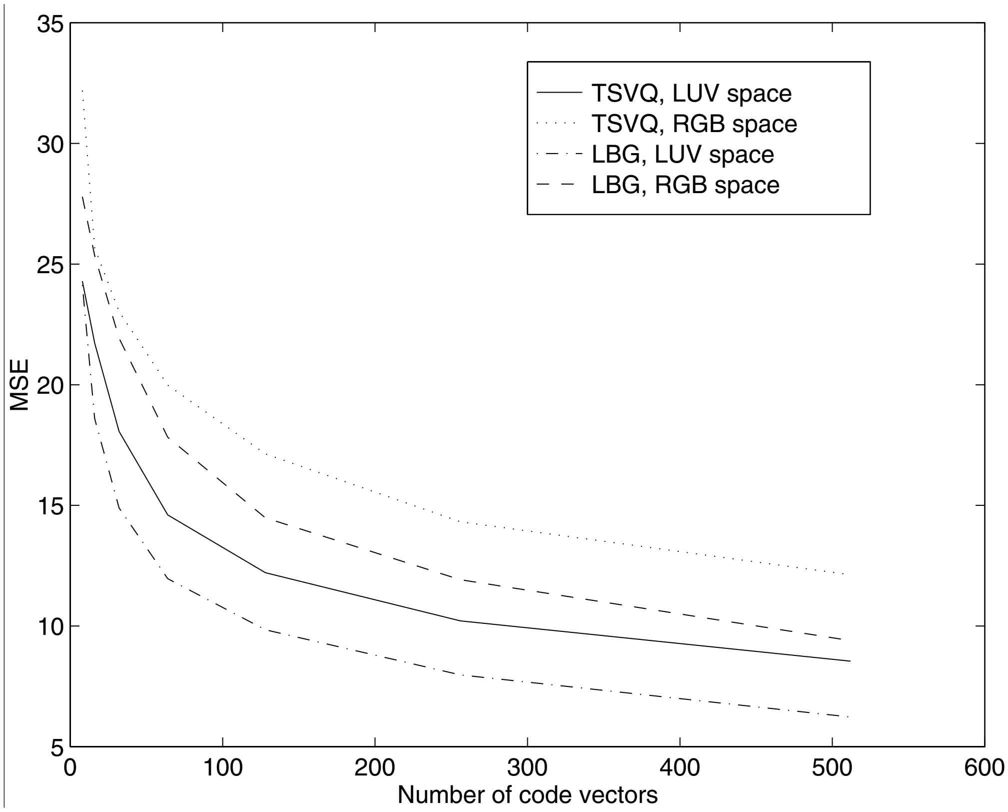

Figure 2. Performance of TSVQ and full-search (LBG) VQ in LUV and RGB spaces ("stained_glass" image)

V. Experimental Results

Vector Quantizer Design

We compare the performance of vector quantizers designed in the RGB and CIELUV spaces. Note that spaces such as YUV and XYZ are obtained though linear transformation from the RGB space, they will have similar results to that of the RGB space. The distance measures used in codebook training and nearest color assignment is the Euclidean distances defined in their respective color space. We use the MSE measure used to evaluate the performances by calculating the average Euclidean distance between the original colors and quantized colors in the LUV space to conform with the human perception of color difference. For the training process, we use 20 training images with medium to high color saturation. The purpose of the saturation choice is to cover the whole color gamut in our color image database. Each image is of size 192×128 with 24 bits of RGB color. A special VQ initialization algorithm[4] is used in LBG VQ and TSVQ design since it has the property that the code vectors tend to separate from each other as much as possible. Two codebooks are trained, one in RGB space, and the other in LUV space.

In the test cases, we includes some images included in the training set and some outside the training set are tested. The quantizer performance in terms of MSE with respect to one image in the training set ("stained_glass") and one outside the training set ("horse") are plotted in Figure 2 and 3. We see from these two figures that the LUV quantizer outperforms the RGB quantizer by about 20%-40%. As for the visual performance, the LUV quantizer has higher fidelity in preserving the original color. However, it also have more severe pseudo-boundary effect when the number of color is very small. The reason is that the mapping in VQ tries to assign the closest color from the color palette to the specific pixel. Since the LUV quantizer reproduces the actual color more faithfully in the perceptual sense, the gradual color change of a surface will result in a pseudo boundary on that surface. However, since the RGB quantizer did not strictly minimize the color difference perceptually, it acts more like a dithering techniques[8] can make the LUV quantized results be comparably pleasant as that of the RGB quantizer, while still having the advantage of color fidelity.

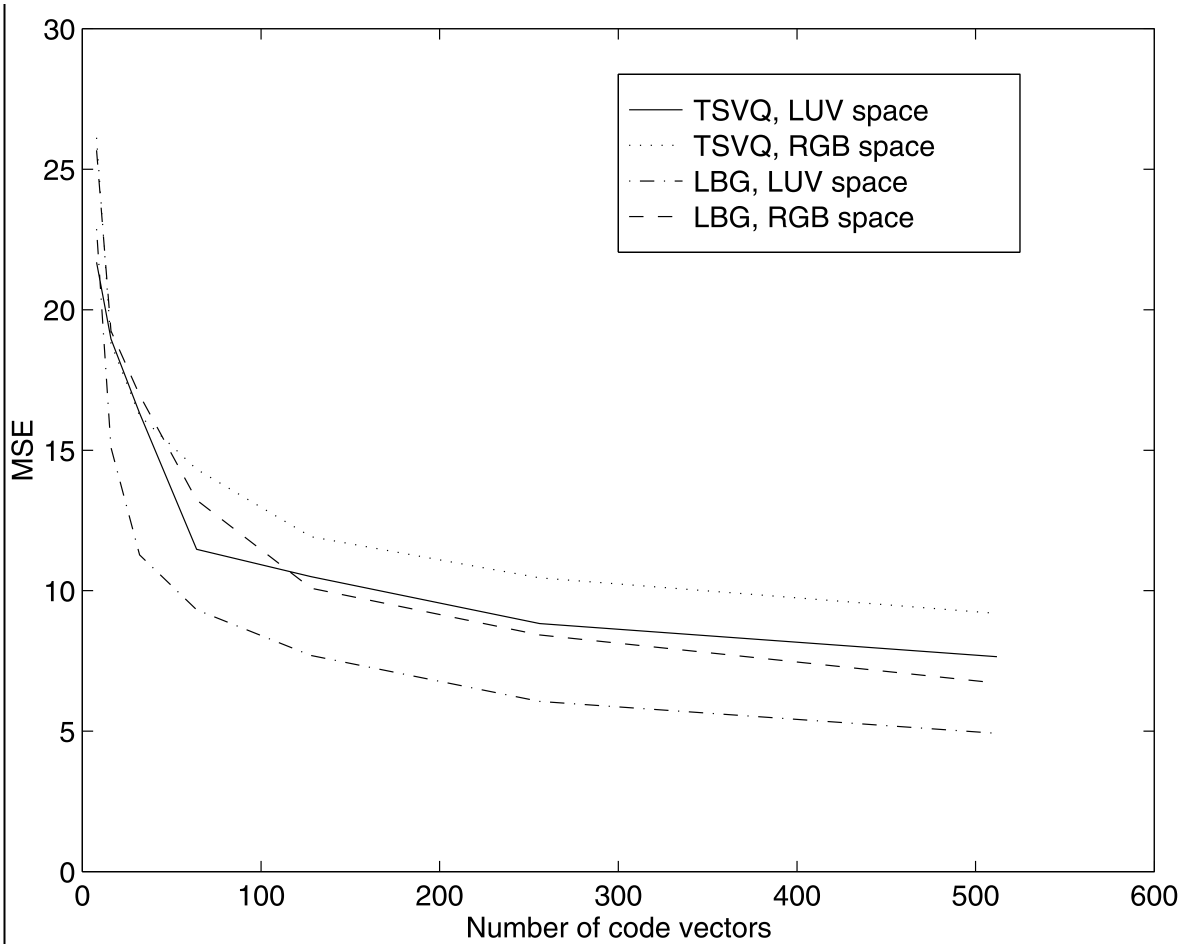

Figure 3. Performance of TSVQ and LBG VQ in LUV and RGB spaces ("horse" image)

Figure 4 and 5 show the performances of several different vector quantizers for these two images. From Figure 1, we see that it is impossible to partition the LUV space with orthogonal axes into cubes. Therefore, we only compare their performances in the RGB color space. For the statistic distribution used in PVQ design, the same 20-image database is used. We see that the LBG (full-search) VQ is the best among all as predicted, and TSVQ has a pretty good performance when compared to uniform and Product VQ, where the latter two have about the same performance. From Figure 2 to Figure 5, we see that whether the test image is in the training set or not, the performances of the vector quantizers are about the same. Therefore, a "universal" codebook for color quantization is feasible in practice, which will reduce the computational complexity of training.

Image Retrieval

We compare the retrieval efficiencies for four color quantization schemes: uniform quantization in RGB, product VQ in RGB, Tree-structured VQ in RGB, and Tree-structure VQ in LUV. "sunsets" and "stained_glasses" were used as query images. There are 8 "sunsets" images and 5 "stained_glasses" in our experimental database which consists of 2119 images in total. The contents of the image database include natural scenes, people, animals, plants and architectures.

The retrieval result for a query was a list of images, ranked by their similarities to the query image. The ideal result would be that all "sunsets" were ranked at the top 8 positions, and all "stained_glasses" images ranked at the top 5 positions, for the respective queries. To evaluate the retrieval performance, we list the number of "sunsets" images ranked at the first 8 positions and the number of "stained_glass" images in the first 5 positions of the retrieval list. They are shown in Table 1 to Table 4.

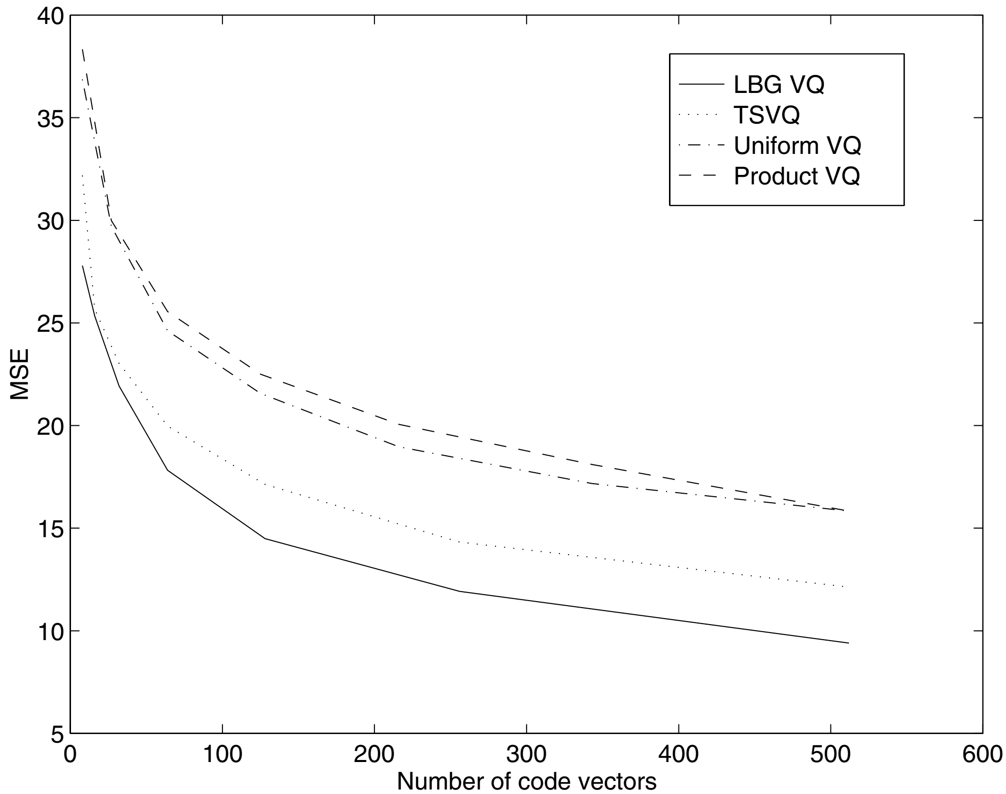

Figure 4. Performance comparison between various vector quantizers in the RGB space ("stained_glass" image)

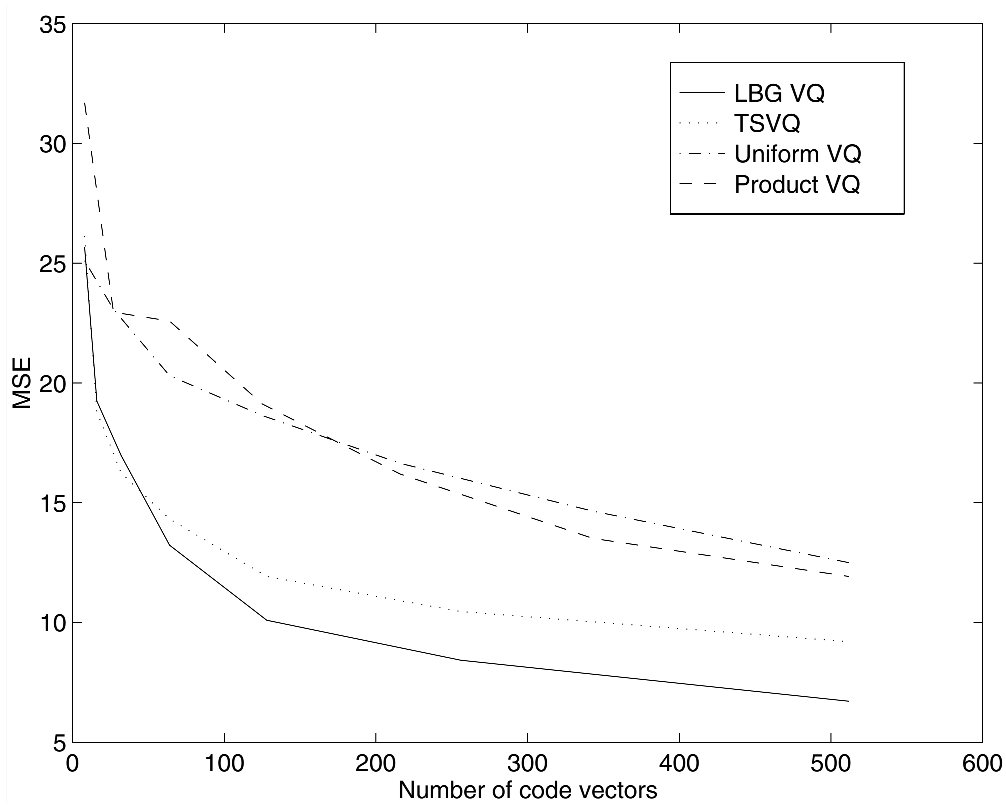

Figure 5. Performance comparison between various vector quantizers in the RGB space ("horse" image)

Table 1. Retrieval Results of Uniform Quantization in the RGB Space

Uniform Quantizer	sunsets	stained glass
(2,2,2)	2	1
(3,3,3)	4	1
(4,4,4)	3	4
(5,5,5)	3	4
(6,6,6)	4	4
(7,7,7)	4	5

Table 2. Retrieval Results of Product VQ in the RGB Space

Product VQ	sunsets	stainedglass
(2,2,2)	2	1
(3,3,3)	3	3
(4,4,4)	3	4
(5,5,5)	4	4
(6,6,6)	3	4
(7,7,7)	4	5

Table 3. Retrieval Results of Tree-structured VQ in the RGB Space

TSVQ	sunsets	stained glass
8 bins	1	3
16 bins	2	3
32 bins	3	5
64 bins	3	5
128 bins	3	5
256 bins	3	5

Table 4. Retrieval Eesults of Rree-structured VQ in the LUV Space

TSVQ	sunsets	stained_glass
8 bins	2	3
16 bins	3	3
32 bins	5	4
64 bins	4	5
128 bins	4	5
256 bins	5	5

Cartoon rendering

We have rendered many cartoon images ranging from simple Mickey Mouse head to textured grasslands. We choose RGB, LUV and LAB color spaces and perform TSVQ with three different initialization, i.e. Monte-Carlo, maximum distance and splitting. These images are rendering with 16 to 256 different colors. According to our experimental result, TSVQ on LUV color space with the maximum distance initialization gives best performance in term of both MSE and visual measurement. In general, 16 colors are sufficient for the recognition purpose. One can easily understand every objects in the scene. Strokes within a region are distinctive and have the same cosmetic effect. If the image is rendered by 256 colors, one can barely tell the color difference, the color looks close to the original color and the changing of colors within a region is smooth. Overall, we think the TSVQ is a simple yet very powerful rendering technique.

Conclusion

We have investigated the color quantization schemes in different color spaces, and their effects on the performance when applied to color image retrieval and cartoon rendering. For image retrieval, the retrieval rate is higher when the granulate of the quantization gets smaller. To reduce the computational complexity, lower resolution features can be used first to exclude irrelevant images. Retrieval using TSVQ in the LUV space performs almost the same as that of in the RGB space, although the quantization error of individual image in LUV space is smaller. This is because retrieval by global histograms is not so sensitive to the scale linearity of color. Experimental results also show that although product VQ is much less complex than TSVQ, it results in a satisfying retrieval performance in the RGB space. For cartoon rendering, we use TSVQ to perform a simple but efficient coding scheme which keeps both visual and entertainment elements of the image. From the design and experiments of these hierarchical color quantizers, we see that they offer good performances with low computational complexity and are very useful for many image processing applications.

References

1. D. L. MacAdam, Sources of Color Science, Cambridge, Mass: MIT Press, 1970.
2. A. K. Jain, "Color distance and geodesics in color 3 space," *J. Opt. Soc. Am.* vol. **62**, 1972, pp. 1287-1291.
3. M. R. Pointer, "A comparison of the CIE 1976 color spaces," *Color Research and Application*, vol. **6**, pp. 108-118.
4. I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang, "Fast tree-structured nearest neighbor encoding for vector quantization," *IEEE Trans. on Image Processing*, vol. **5**, no. 2, Feb. 1997, pp. 398-404.
5. Xia Wan and C.-C. J. Kuo, "Color distribution analysis and Quantization for Image Retrieval," *Storage and Retrieval for Still Image and Video Databases IV*, vol. **2670**, SPIE, Feb. 1996, pp. 8-16.
6. H. J. Zhang, Y. Gong, C. Y. Low, and S. W. Smoliar, "Image retrieval based on color features: an evaluation study," *Digital Image and Archiving Systems*, vol. **2606**, SPIE, Oct. 1995, pp. 212-220.
7. M. J. Swain and D. H. Ballard, "Color indexing", *International Journal of Computer Vision*, vol. **7**, no. 1, 1991, pp. 11-32.
8. M. T. Orchard and C. A. Bouman, "Color quantization of images," *IEEE Trans. On Signal Processing*, vol. **39**, no. 12, Dec. 1991, pp. 2677-2690.