Color Quantization by Hierarchical Octa-Partition in RGB Color Space

Chih-Hung Lee[1], Hao-ying Lu[2] and Ji-Hwei Horng[3]

Department of Electronic Engineering, National Quemoy University, Kinmen, Taiwan
[1]peterlee@email.nqu.edu.tw, [2]hylu@email.nqu.edu.tw, [3]horng@email.nqu.edu.tw

Abstract

Digital color images are widely used in our daily life, especially for smart-phone users. However, the storage and processing of the digital images consume a lot of hardware resources. To reduce the data amount of a true-color image, the indexed image is one common solution. In transforming a true-color image into an indexed image, the true-color space should be quantized into 256 colors to meet the usual size of a color-map. In this paper, we propose a new method to analyze an image' s color distribution and produce its color-map for transforming to an indexed image.

In most natural images, the color distribution of the image pixels is highly concentrated in some restricted regions. Equally-spaced quantization, of course, does not produce satisfactory results. We propose a new method that iteratively partitions the color space cube into eight equally sized sub-cubes. However, in doing so, we apply a pre-defined criterion to decide whether each sub-cube will be further partitioned. The decision criterion is based on the population size of the sub-cube. In this way, the regions where the most pixels concentrate in will be partitioned into finer sub-cubes and thus increases their discrimination after quantization.

We applied the new proposed method to several types of true-color images and analyzed the resulting numerical errors. In addition, we utilized the error diffusion dithering technique to improve their visual effect. Satisfactory results were obtained.

Key words: color quantization, RGB color space, indexed image, error diffusion, dithering

Introduction

Due to the rapid development of the portable electronic devices, the communication technology, and the digital image acquisition equipment, digital images are widely applied in our daily life. Although the hardware is also upgraded quickly, we still wish to reduce the data amount of digital images. The storage, processing, and transmission of digital images may cost expensive hardware resources or service fees.

To reduce the image data amount, color indexing is a typical adopted technique. In doing so, color quantization is a required preprocess. Some methods for color quantization have been proposed [1~3]. In the image processing toolbox of the popular commercial software MATLAB [4], there are two types of quantization options, the uniform quantization (UQ) and the minimum variance quantization (MVQ).

In the uniform quantization, the RGB color cube is uniformly divided by a predefined tolerance. Then, the empty sub-cubes are deleted. The maximum number of colors and the maximum quantization error can be controlled by the tolerance value. In the minimum variance quantization, the color cube is divided into sub-cubes according to the variance between pixel values. A set of pixels with a small variance from their center pixel are grouped together. The resulting sub-cubes vary in size. Therefore, the resulting image may have a few pixels with rather large quantization errors. In addition, the computation of the minimum variance method is more complicated and thus more time-consuming.

In this paper, we propose a novel method for color quantization. The proposed method hierarchically quantizes the color cube according to the color distribution of the given image. The details are described in the following section. Then, we test and compare our method with the existing methods. Finally, conclusions will be presented.

The Hierarchical Octa-Partition Quantization (HOQ)

The basic idea of our method is to partition the color cube according to the local density of the image pixel distribution. In the base level, the color cube is octa-partitioned into eight equally sized sub-cubes by half-cutting in each direction of the x, y, and z axes. In the next level, we further examine each of the sub-cubes produced in the previous level. If the pixel number in the current cube is larger than a given threshold, it is again octa-partitioned into eight finer cubes. The iterative process proceeds until the pixel number of all cubes are lower than the given threshold. The pseudo code of the algorithm is as follows.

1. input an image A and a threshold of pixel number n_t
2. mark the color of each image pixel in the color cube X
3. set level $l = 1$ and octa-partition the color cube into $X_1^l, X_2^l, X_3^l, ..., X_8^l$.
4. set $k = 0$.
5. for all i, count the number of pixels $n(X_i^l)$ in X_i^l.

    if $n(X_i^l) > n_t$

        partition X_i^l into

ISBN 978-1-5386-4342-6

$X_{k+1}^{l+1}, X_{k+2}^{l+1}, X_{k+3}^{l+1}, \ldots, X_{k+8}^{l+1}$ and $k = k + 8$.

endif

endfor

6. while partition occurs, $l = l + 1$ and goto 4;
   else, delete all empty cubes and terminate.

By applying our algorithm, the resulting color cubes vary in size but are guaranteed to have pixel number less than the given threshold. We hierarchically partition the dense cubes and thus increase their discrimination.

A problem may arise. How do we decide the threshold of pixel number? We can start with a typical value, for example 1000, and then execute the algorithm to find the resulting number of colors. If there are more than 256 colors, increase the threshold; else, decrease the value. In this way, we can find an optimal threshold that produces the largest number of colors below 256.

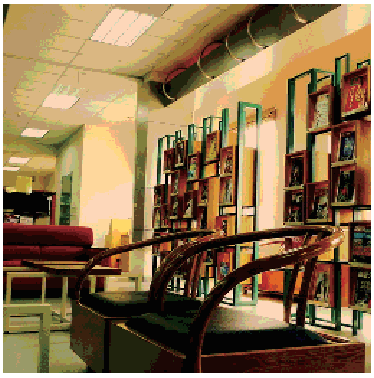
Experimental Results

The proposed method is applied to different images obtained from different situations, including indoor room(Fig. 1) and outdoor buildings(Fig. 2, 3); in the daytime(Fig. 1,2) and in the night(Fig. 3). For easy of comparison, we also provide the quantization results of the methods UQ and MVQ for each test image. Observing the images, our method produces satisfactory results which are comparable with the existing ones in human vision.

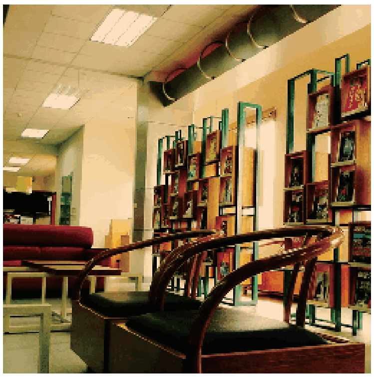
In practical applications, the color quantization is usually applied together with the dithering technique to reduce the contour effect resulting from quantization. Fig. 4 shows the experimental results of these methods applied with dithering technique. In our experiment, we adopt the Floyd－Steinberg dithering technique [5]. Due to the existence of empty spaces in the quantized color cube, the quantization algorithm should be somewhat modified to coordinate with the dithering algorithm. In the Fig. 4, the UQ and MVQ images are produced by applying the MATLAB functions [4]. Again, the performance can hardly be discriminated by the human eyes.

To make a fair comparison of the three methods, we adopt two typical error measuring indexes, the root-mean-square error (e_{rms}) and the peak error (e_{peak}). The errors for the three test images are given in Table I~III. The size of color palette can be assigned precisely in applying the MVQ function. However, the exact sizes in the remainders are determined automatically by the algorithms. We set the palette size of the MVQ equal to our method and adopt the result of the UQ with a slightly larger size. Referring to the tables, the MVQ has excellent performance in the e_{rms} index, and the UQ is better in the e_{peak} index. In all three cases, the performance of our method is in between.

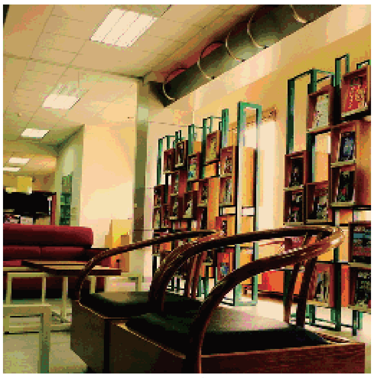
(a)



(b)



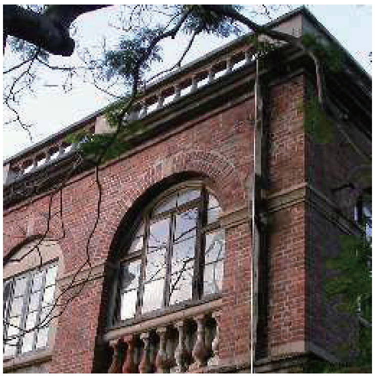
(c)



(d)

Fig. 1 Experimental Results of the three color quantization methods on test image # 1, (a) the original true-color image; (b) the image produced by UQ; (c) the image produced by MVQ; (d) the image produced by the proposed method.

ISBN 978-1-5386-4342-6


(a)


(b)


(c)


(d)

Fig. 2 Experimental Results of the three color quantization methods on test image # 2, (a) the original true-color image; (b) the image produced by UQ; (c) the image produced by MVQ; (d) the image produced by the proposed method.


(a)

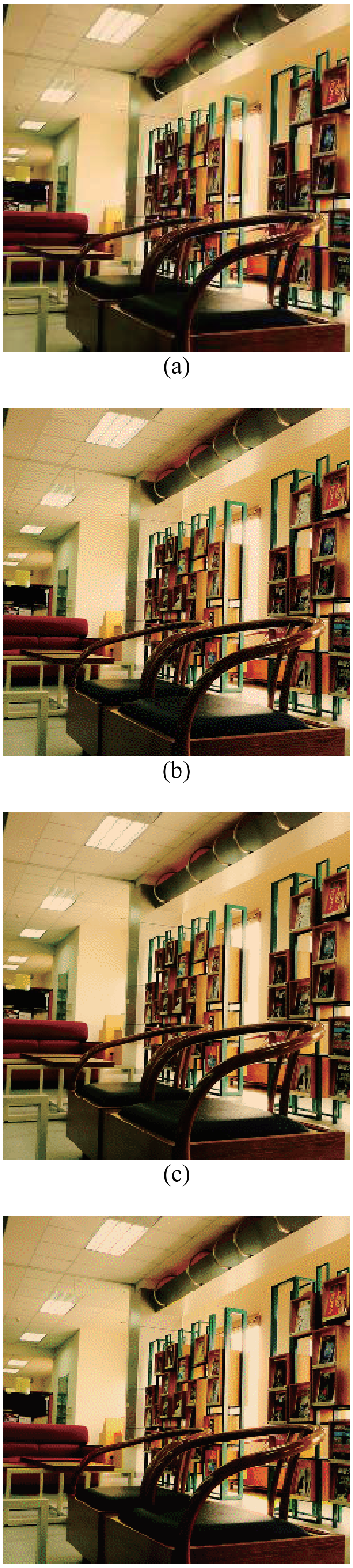

(b)


(c)


(d)

Fig. 3 Experimental Results of the three color quantization methods on test image # 3, (a) the original true-color image; (b) the image produced by UQ; (c) the image produced by MVQ; (d) the image produced by the proposed method.

ISBN 978-1-5386-4342-6



(a)

(b)

(c)

(d)

Fig. 4 Experimental Results of the three color quantization methods combined with dithering on test image # 1, (a) the original true-color image; (b) the image produced by UQ; (c) the image produced by MVQ; (d) the image produced by the proposed method.

TABLE I

NUMERICAL COMPARISON OF THE THREE COLOR QUANTIZATION METHODS ON TEST IMAGE # 1

Method	Size of Color Palette	e_{rms}	e_{peak}
UQ	185 colors	15.0957	27.7128
MVQ	175 colors	8.4688	62.8092
HOQ	175 colors	13.8646	54.8544

TABLE II

NUMERICAL COMPARISON OF THE THREE COLOR QUANTIZATION METHODS ON TEST IMAGE # 2

Method	Size of Color Palette	e_{rms}	e_{peak}
UQ	198 colors	14.5019	24.2487
MVQ	180 colors	7.9647	77.0519
HOQ	180 colors	13.9846	54.8544

TABLE III

NUMERICAL COMPARISON OF THE THREE COLOR QUANTIZATION METHODS ON TEST IMAGE # 3

Method	Size of Color Palette	e_{rms}	e_{peak}
UQ	171 colors	16.7322	27.7128
MVQ	151 colors	7.9076	71.0634
HOQ	151 colors	14.0076	53.7122

Conclusions

In this paper, we propose a novel color quantization method whose performance is comparable with the popular commercial software both in visual effects and in mathematical indexes. In addition, the proposed method is simple and easy to implement. Experimental results and their error measures are given.

References

[1] Celal Ozturk, Emrah Hancer, and Dervis Karaboga, Color Image Quantization: A Short Review and an Application with Artificial Bee Colony Algorithm, Vol. 25, No. 3, pp. 485 – 503, ***INFORMATICA***, 2014.

[2] P. Heckbert, Color image quantization for frame buffer display, ***Proceedings of 9th Annual Conference on Computer Graphics and Interactive Techniques***, Boston, Massachusetts, United States, pp. 297-307, 1982.

[3] P. Heckbert, Color image quantization for frame buffer display, ***COMPUTER GRAPHICS***, 16(3), 297-307, 1982.

[4] https://www.***mathworks***.com/help/images/reduce-the-number-of -colors-in-an- image.html.

[5] R.W. Floyd, L. Steinberg, An adaptive algorithm for spatial grey scale. ***Proceedings of the Society of Information Display 17***, 75-77 , 1976.

ISBN 978-1-5386-4342-6