Comparison and Optimization of Methods of Color Image Quantization

Jean-Pierre BRAQUELAIRE and Luc BRUN
LaBRI – 351, cours de la libération
F-33405 Talence CEDEX

Abstract

Color image quantization is the process of reducing the number of colors in a digital color image has been widely studied for the last fifteen years. In this paper the different steps of clustering methods are studied. The methods are compared step by step and some optimizations of the algorithms and data structures are given. A new color space called $H1H2H3$ is introduced which improves quantization heuristics. A low-cost quantization scheme is proposed.

Keywords

Color quantization, clustering, color spaces, human vision, data structures.

Contact with authors

Jean-Pierre BRAQUELAIRE
Phone: +33 56.84.60.91 – E-mail: `Jean-Pierre.Braquelaire@LaBRI.U-Bordeaux.Fr`
Luc BRUN
Phone: +33 56.84.24.31 – E-mail: `Luc.Brun@LaBRI.U-Bordeaux.Fr`

Fax Number : +33 56.84.66.69

Unité associée au CNRS n°1304 – Université Bordeaux I

I. Introduction

Color image quantization is the process used to reduce the number of colors in a digital color image. This process may be used to compress the image information or to display a 24-bits per pixel image on a monitor which can only display a reduced set of colors. If the aim of the color quantization algorithm is to display images, the visual difference between the quantized images and the original ones has to be minimal. Let us consider a color image $\boldsymbol{I}$ and let us denote by $\mathcal{C}$ the set of its colors and by M the cardinality of $\mathcal{C}$. The quantization of $\boldsymbol{I}$ into K colors, with $K < M$ (and usually $K << M$) consists in selecting a set of K *representative colors* and replacing the color of each pixel of the original image by the *"closest"* representative color. The quantization process may be uniform or adaptable. In the case of a uniform quantization [10], the set of representative colors is preselected and remains the same for all images. In this work, we consider adaptive quantization: the set of representative colors $\{c_1, c_2, ...c_K\}$ is built as a function of the image to be quantized. The result of this construction is a function $\boldsymbol{Q} : \mathcal{C} \longrightarrow \{c_1, c_2, ...c_K\}$ associating each color of the image with its nearest representative color. The function $\boldsymbol{Q}$ defines a partition of the image color set into a set of K subsets called **clusters**. Thus we have $\mathcal{C} = \bigsqcup_{i=1}^{K} \boldsymbol{C}_i$, with $\boldsymbol{C}_i = \boldsymbol{Q}^{-1}(c_i)$.

In order to minimize the error between quantized images and original ones, the set of colors that the function $\boldsymbol{Q}$ associates with a color c_i has to be close to c_i. The error associated with a color c_i can be defined as :

$$\boldsymbol{E}(\boldsymbol{C}_i) = \sum_{c \in C_i} f(c) \parallel c - c_i \parallel^2$$

where $f(c)$ is the number of pixels with color c in the image.

A well-known mathematical result shows that $\boldsymbol{E}(\boldsymbol{C}_i)$ is minimal when c_i is the centroid of $\boldsymbol{C}_i$. Then $\boldsymbol{E}(\boldsymbol{C}_i)$ measures the homogeneity of cluster $\boldsymbol{C}_i$. The total error $\boldsymbol{SE}(\boldsymbol{C})$ between a quantized image and its original can then be defined as the sum of the errors on all clusters.

$$\boldsymbol{SE}(\boldsymbol{C}) = \sum_{i=1}^{K} \boldsymbol{E}(\boldsymbol{C}_i) \tag{1}$$

It is known that the problem of finding a partition of the image color set that minimizes equation 1 is NP-complete [13]. For this reason different techniques have been used to compute an approximation of the optimal partition. Using such approximations, iterative optimization procedures such as the *dynamic clusters method* [4] may improve the initial solution.

The heuristics used to find an approximation of the optimal solution can be divided into two main approaches: a bottom-up approach and a top-down one. The general principle of bottom-up approaches is to select K colors of the image to initialize the K clusters. The other colors of the image are then sequentially read and merged into the clusters according to different heuristics [5], [15]. Since these heuristics have to be applied to the pixels of the image, they are often very simple in order to keep the algorithms efficient. Moreover the final clusters are greatly influenced by the colors used to initialize the clusters and by the order of the scan.

The top-down approach has been much more investigated [6], [11], [9], [13], [14], [1]. Here the principle is to initialize a cluster containing all the colors of the image and recursively split it into two sub-clusters until the desired number of clusters is obtained. This process creates a binary tree resulting from the splits. The heuristics used to choose a cluster and to split it into two sub-clusters try to minimize the error defined by equation 1. In order to simplify the construction, each split is usually done along a plane separating a cluster into two sub-clusters. We call this plane the **splitting plane**. The axis orthogonal to this plane is called the **splitting axis**. The splitting process can be then decomposed into four sub-processes:

1. Select the next cluster to be split.
2. Select a set of possible splitting axes.
3. Use a criterion to select the splitting axis.
4. Determine the position of the splitting plane along the splitting axis.

The heuristics used by our algorithm are displayed in Table IV. This Table also resumes the heuristics used by the algorithms of Heckbert [6], Wan et al. [11] and Wu [13]. In this paper we compare various top-down heuristics and we propose optimizations for some heuristics. In Section II we propose the use of a new color space to improve the selection of the splitting axis and to add the benefit of a metric closer to human vision than the RGB color space. In Section III we give an improvement of an algorithm to select the splitting plane. In Section IV we describe an optimal encoding of the set of colors of the image to be quantified that allows efficient computation of the statistical parameters used during each step of the clustering process. Finally, we compare our method to Wu's method in section V.

II. Selection of the color space

In this section we suppose that an heuristic has been used to determine the cluster to be split. In our algorithm this cluster is the one with the greatest squared error. The next heuristic concerns the choice of a set of possible splitting axes. Many authors [13], [11], [9] split each cluster orthogonally to its major axis. It can be shown (see [3]) that this heuristic, called the *major axis heuristic*, can lead to poorer results than those obtained by the *fixed axis heuristic*, which splits clusters orthogonally to their coordinate axis of greatest variance. This fact is confirmed by experiments summarized in Table I, where the quantization algorithm used the $I_1I_2I_3$ color space defined by Otha [8]. The fixed axis heuristic usually gives worse results, but the difference may be small and the use of fixed directions allows us both to avoid computing the covariance matrix of the cluster and to manipulate more regular data. We thus decided to split each cluster orthogonally to its coordinate axis of greatest variance.

The problem is then to determine a good color space and thus a good set of coordinate axes for color quantization algorithms. Experimental results showed us that the $I_1I_2I_3$ color space gives bad visual results on some images. The reason seems to be that this color space has a metric which does not correspond to human vision. Thus we have experimented with other color spaces closer to human vision than RGB or $I_1I_2I_3$. These visual experiments led us to choose the $H_1H_2H_3$ color space defined by:

$$(H_1 = R + G, H_2 = R - G, H_3 = B - \frac{R+G}{2})$$

This model is derived from Faugera's theory of human vision, which incorporates the widely-established notions of *trichromaticity* and *color opponency* theory [2], [7]. According to this theory the color information stored by the retinal photoreceptors is transmitted to the brain along three channels. The achromatic channel corresponds to the sum of long and medium wavelengths, respectively picked up by the M and L cones. In our color space this channel is approximated by the axis $H_1 = R + G$. The two other channels are called the *red-green* and *blue-yellow* channels. They are respectively approximated by the axes H_2 and H_3.

Experiments summarized in Table III show that the difference between the use of fixed axis and major axis quantization algorithms with the $H_1H_2H_3$ color space is only 2% for a quantization using 16 colors and 7% for a quantization into 256 colors. Moreover, experiments summarized on Table II show that splits orthogonal to the coordinate axes improve the execution times of quantization algorithms by about 170% for a quantization into 16 colors and by 220% for a quantization into 256 colors.

Note that other color spaces may obtain equal or better results. However, visual experiments performed on the RGB, $I_1I_2I_3$, YIQ and $L^*u^*v^*$ color spaces show that the use of these color spaces does not clearly induce an improvement of the visual quality of final images. Moreover, these color spaces require a greater computational cost than the $H_1H_2H_3$ color space.

III. Determination of the splitting plane

In this section we suppose that the splitting axis is determined, and we focus on determining the position of the splitting plane along the splitting axis. Let us suppose that the cluster $\boldsymbol{C}$ has to be split along the axis A. Let a_1 and a_2 be the extremities of $\boldsymbol{C}$ along the axis A, see Fig. 1. Let t be the position on the axis A of the splitting plane and $\boldsymbol{C}_t$ and $\boldsymbol{C}-\boldsymbol{C}_t$ the sub-clusters resulting from the splitting. The problem of determining of the best position of the splitting plane can be formulated as follows: find a value of $t \in [a_1, a_2]$, that minimizes the quadratic error associated with the splitting of $\boldsymbol{C}$ into $\boldsymbol{C}_t$ and $(\boldsymbol{C} - \boldsymbol{C}_t)$. It can be shown [13], [3] that the best position of the plane is such that the sum of the squared errors of both split clusters is minimal. We denote by $\boldsymbol{E}'_t(\boldsymbol{C})$ this sum. We have:

$$\boldsymbol{E}'_t(\boldsymbol{C}) = \boldsymbol{E}(\boldsymbol{C}_t) + \boldsymbol{E}(\boldsymbol{C} - \boldsymbol{C}_t) \tag{2}$$

The optimal value t_{opt} is the value of t that minimizes $\boldsymbol{E}'_t(\boldsymbol{C})$. This value is determined by computing $\boldsymbol{E}'_t(\boldsymbol{C})$ for each possible value of t of the sequence of t_i. In this section we give a new formulation of $\boldsymbol{E}'_t(\boldsymbol{C})$ that allows an optimization of the computation of t_{opt} by reducing the set of values of t on which $\boldsymbol{E}'_t(\boldsymbol{C})$ is computed.

Lemma 1: According to the previously defined notations, Equation (2) is equivalent to $\boldsymbol{E}'_t(\boldsymbol{C}) = \boldsymbol{E}(\boldsymbol{C}) - |\boldsymbol{C}|g(t)$ with $g(t) = \frac{\delta(t)}{1-\delta(t)}\|\mu(t) - m\|^2$, and the optimal value of t is $t_{opt} = \underset{t \in [a_1, a_2]}{\text{Arg max}}\, g(t)$, where $\mu(t)$ is the mean of cluster $\boldsymbol{C}_t$, m the mean of cluster $\boldsymbol{C}$, $|\boldsymbol{C}|$ denotes the cardinal of cluster $\boldsymbol{C}$ and $\delta(t)$ is defined by $\delta(t) = \frac{|\boldsymbol{C}_t|}{|\boldsymbol{C}|}$.

The reader may refer to [3] for a proof of this result.

This new formulation has two advantages. The first one is related to the order of size of the quantities involved in the computation of $\boldsymbol{E}'_t(\boldsymbol{C})$, which is smaller when using the equations of Lemma 1. Thus rounding errors that occur when using formula (2) are thus avoided here. The second one concerns an optimization of the computation given by the following theorem.

Theorem 1: According to the previously defined notations,

$$\exists \Lambda \in \mathbb{R}^n, \alpha \in \mathbb{R} \ / \ \forall t \in [a_1, a_2] \quad L(t) \leq g(t) \leq U(t),$$

where:

$$U(t) = \delta(t)(1 - \delta(t))\|\Lambda\|^2 \ \text{ and } L(t) = \chi_{[a_1,\beta]}\frac{\delta(t)}{1 - \delta(t)}\alpha + \chi_{]\beta,a_2]}\frac{1 - \delta(t)}{\delta(t)}\alpha$$

where χ is the characteristic function.

This theorem is illustrated in figure 2. A proof of this result may be found in [3].

The vector Λ is a vector whose norm is greaterthan $\|c - m\|$ for all c in $\boldsymbol{C}$. In our application we set $\Lambda_i, i = 1, \ldots 3$ to the length of the cluster along the axis X_i.

The function $g(t)$ is bounded above by parabola U and bounded below by hyperbola L. Thanks to the parabola U we know that we cannot find the maximum of $g(t)$ for δ greater than δ_{max} (see Fig.2). We need not therefore to traverse the whole interval. It is easy to verify that the projection of the maximum of $g(t)$ gives a point δ_{max} greater than or equal to $\frac{1}{2}$ (see Fig.2). Thus, we must at least traverse the interval $[a_1, a_2]$ from its minimum a_1 to a point t such that $\delta(t)$ is equal to $\frac{1}{2}$. The data distribution is a parameter of the function δ. Thus the improvement of the search varies from one image to another. Each iteration of the algorithm treats a new cluster, and thus for a given image the gain is different from an iteration to another one. Experiments have shown us that the mean reduction of intervals with a quantization to 16 colors is about 10%.

Note that, since the functions U and L reach their maximum when δ is equal to $\frac{1}{2}$, the maximum of $g(t)$ is reached for a value of t such that $\delta(t)$ is about $\frac{1}{2}$. This value $\delta = \frac{1}{2}$ corresponds to the median position of the splitting plane selected by the median cut algorithm proposed by Heckbert.

IV. Optimal encoding of the image color set

We saw in Section II that each cluster is split along its axis of greatest variance. These variances and the optimal value t_{opt} defined in Lemma 1 require the computation of the sets D_α defined by $D_\alpha = \{c \in \boldsymbol{C} \ / \ c \bullet \vec{A} = \alpha\}$, where $\bullet$ denotes the scalar product and $c \bullet \vec{A}$ denotes the orthogonal projection of c onto the splitting axis A. The efficiency of the algorithm highly depends on its efficiency to access the colors of a given set D_α.

The data structure used by Wu [14] stores all colors of the original image in an array. This method does not allow quick access to the data wich have to be sorted at each step of the algorithm. Using fixed axes allows us to avoid this last disadvantage.

We remark that the computation of the sets D_α can be improved by sorting the data along each axis. The coordinates of all the colors of the image are stored in an array $\boldsymbol{V}$ by mean of a hash table. This hash table allows us to efficiently retrieve colors previously stored. If we suppose that the hash table allows us to retrieve a given color in constant time, the complexity of this first step is equal to $\mathcal{O}(S)$, where S denotes the size of the image. For each color, the number of pixels of this color in the image is also stored in $\boldsymbol{V}$. The array $\boldsymbol{V}$ is sorted along each color coordinate, say c_1, c_2 and c_3, thanks to three other arrays of indices $\boldsymbol{ind}_{c1}$, $\boldsymbol{ind}_{c2}$ and $\boldsymbol{ind}_{c3}$. For instance the array $\boldsymbol{ind}_{c1}$ is composed of ranges of consecutive indices of colors having the same c_1 value in $\boldsymbol{V}$. Entries of c_1 (respectively c_2 and c_3) having the same value in $\boldsymbol{ind}_{c1}$ (respectively in $\boldsymbol{ind}_{c2}$ and $\boldsymbol{ind}_{c3}$) are sorted along c_2 (respectively c_1 and c_1). Thus the arrays $\boldsymbol{ind}_{c1}$, $\boldsymbol{ind}_{c2}$ and $\boldsymbol{ind}_{c3}$ may be generated by three quicksorts with the appropriate comparison. Thus the additional complexity induced by the generation of the arrays $\boldsymbol{ind}_{c1}$, $\boldsymbol{ind}_{c2}$ and $\boldsymbol{ind}_{c3}$ is equal to $\mathcal{O}(3N \log N)$, where N denotes the number of different colors contained in the image.

For each $\boldsymbol{ind}_{ci}$, array another array of indices denoted by $\boldsymbol{entry}_{ci}$ encodes the beginning of each range of $\boldsymbol{ind}_{ci}$. This last operation may be done by traversing each array $\boldsymbol{ind}_{ci}$. The complexity of this operation is thus equal to $\mathcal{O}(3N)$.

The data structures $\boldsymbol{V}$, $\boldsymbol{ind}_{c1}$, and $\boldsymbol{entry}_{c1}$ are summarized on Fig. 3. Note that for every value of j in $\{1,2,3\}$, for any k in $[Min_j, Max_j]$, and for any $l \geq \boldsymbol{entry}_{cj}[k]$ we have $\boldsymbol{V}[\boldsymbol{ind}_{cj}[l]].c_j \geq k$. Thus for a given value k, the array $\boldsymbol{entry}_{cj}$ provides a direct access to the first color having a c_j coordinate greater than k. This last property is used to determine the sets D_α. The problem can be formulated as follows:

Given a cluster $\boldsymbol{C}$ and a real number $\alpha \in [m_j, M_j[$, the set D_α is equal to the colors defined by $c_i = \alpha$, $c_j \in [m_j, M_j[$, and $c_k \in [m_k, M_k[$, where $[m_i, M_i[\times[m_j, M_j[\times[m_k, M_k[$ defines cluster $\boldsymbol{C}$. Without loss of generality we can assume that $i = 1$, $j = 2$ and $k = 3$. The computation of the moments is then decomposed into three steps:

1. Find the set $\boldsymbol{C}^1$ of colors with $c_1 = \alpha$ thanks to $\boldsymbol{entry}_{c1}$. The set $\boldsymbol{C}^1$ is stored sequentially in $\boldsymbol{ind}_{c1}$.
2. Search in $\boldsymbol{C}^1$ the set of indices $\boldsymbol{C}^2$ satisfying $m_2 \leq V[j].c_2 < M_2$. Since $\boldsymbol{ind}_{c1}$ is a sorted array, the set $\boldsymbol{C}^2$ can be found with two binary searches of m_2 and M_2. The set $\boldsymbol{C}^2$ is stored sequentially in $\boldsymbol{ind}_{c1}$.
3. Traverse $\boldsymbol{C}^2$ and process for the indices verifying $m_3 \leq V[j].c_3 < M_3$.

To sum up, this data structure presents four advantages:

1. Only the colors of the image are stored and processed.
2. The data are efficiently accessed.
3. The data structure is convenient for any color space.

V. Conclusion

In this paper we have presented a new top-down quantization method based on the new color space $H_1H_2H_3$. This method splits each cluster along the axis H_1, H_2 or H_3 of greatest variance. The split of the cluster is improved thanks to theoretical results which generalize some results obtained by Wong [12] for monochrome image quantization. Finally a new data structure is proposed which allows us to manipulate the 3D histogram of an image in an efficient way.

According to experimental results summarized in Table III and Table II, the proposed method can be seen as an intermediate method between a fast but poor method such as the median cut, and better but slower methods such as Wu's one that splits the clusters according to their major axis. Most of the time this intermediate method produces quantized images with a quality almost equal to better but slower methods.

References

[1] Raja Balasubramanian, Charles A. Bouman, and Jan P. Allebach. Sequential scalar quantization of color images. *Journal of Electronic Imaging*, 3(1):45–59, January 1994.

[2] Robert M. Boynton. *Human Color Vision*. Optical Society of America, 1992.

[3] Luc Brun. *Segmentation d'images couleur à base Topologique*. PhD thesis, Université Bordeaux I, 351 cours de la Libération 33405 Talence, December 1996.

[4] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes : La méthode des nuées dynamiques. *INRIA, revue de statistique appliquée*, 19(2):19–33, 1971.

[5] Michael Gervautz and Werner Purgathofer. *A simple method for color quantization: Octree quantization*. Glassner editor, 1990.

[6] Paul Heckbert. Color image quantization for frame buffer display. In *Proceedings of SIG-GRAPH'82*, pages 297–307, 1982.

[7] M.D. Levine. *Vision in man and machine*. Mc Graw-Hill Book Compagny, 1985.

[8] Yu-Ichi Ohta, Takeo Kanade, and Toshiyuki Sakai. Color information for region segmentation. *Computer Vision, Graphics, and Image Processing*, 13:222–241, 1980.

[9] Michael T. Orchard and Charles A. Bouman. Color quantization of images. *IEEE transactions on signal processing*, 39(12):2677–2690, 1991.

[10] Alan Paeth. Mapping rgb triples onto 16 distinct values. In James Arvo, editor, *Graphics Gems II*, volume II, chapter 3, pages 143–146. Academic Press, 91.

[11] S.J. Wan, S.K.M Wong, and P. Prusinkiewicz. An algorithm for multidimensional data clustering. *ACM Trans. Math. Softw.*, 14(4):153–162, 1988.

[12] S.K.M. Wong, S.J. Wan, and P. Prusinkiewicz. Monochrome image quantization. In *Canadian conference on electrical and computer engineering*, pages 17–20, September 1989.

[13] Xiaolin Wu and K. Zhang. A better tree-structured vector quantizer. In *Proceedings of the IEEE Data Compression Conference*, pages 392–401. IEEE Computer Society Press, 1991.

[14] Xiaoling Wu. Color quantization by dynamic programing and principal analysis. *ACM Transaction on Graphics*, 11(4):348–372, 1992.

[15] Zhigang Xiang and Gregory Joy. Color image quantization by agglomerative clustering. *IEEE Computer Graphics and Applications*, 272-17-16:44–48, 1994.

Appendix

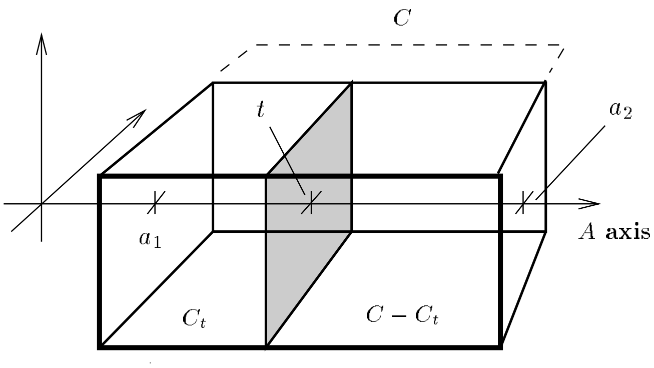

Fig. 1. *The selected cluster* $\boldsymbol{C}$ *is split along a plane orthogonal to the selected axis* A *and positioned at the coordinate* t *on the axis* A.

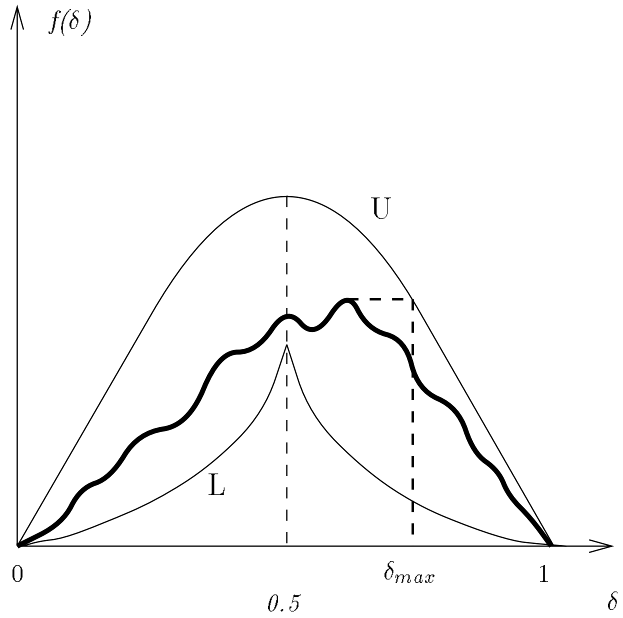

Fig. 2. *$g(t)$ bounded below by an hyperbola and above by a parabola.*

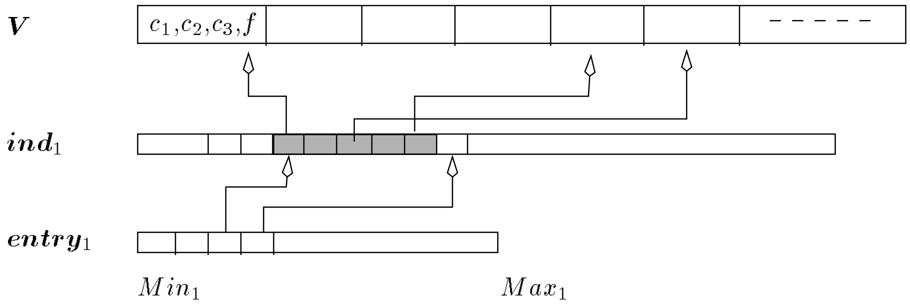

Fig. 3. *The set of colors is sorted along each of the three components*

# of colors	Zelda	Lenna	Flowers	Anemone	Mandrill	Stones	Means
16	-8 %	0 %	8 %	3 %	0 %	4 %	2 %
256	3 %	16 %	6 %	2 %	2 %	-7 %	4 %

TABLE I

This Table summarized quantization experiments showing that both fixed axis heuristic and major axis heuristic may produce comparable results when using an appropriate color space (here the $I_1 I_2 I_3$ color space). In some cases major axis heuristic give worse results than fixed axis heuristic. The percentages given in the table are obtained by the formula $\frac{FA-MA}{FA}$, where FA represents the squared error obtained by the fixed axis quantization algorithm and MA the squared error obtained by the major axis quantization algorithm.

	# of colors	Zelda	Lenna	Flowers	Anemone	Mandrill	Means
(1)	16	42	29	43	1'13	36	44
	256	52	36	49	1'26	41	52
(2)	16	1'06	47	1'18	2'00	1'00	1'14
	256	1'41	1'15	2'06	3'05	1'34	1'56

TABLE II

This table shows the time spent by two algorithms. The first one (1) cuts clusters orthogonally to the coordinate axis of greatest variance. The second one (2) cuts clusters orthogonally to their major axis. Both algorithms use the $H_1H_2H_3$ color space.

# of colors	Zelda	Lenna	Flowers	Anemone	Mandrill	Means
16	2 %	1 %	2%	4%	4%	2%
256	14%	9%	5%	7%	5%	7%

TABLE III

This table compares the squared error obtained by two quantization algorithms with the $H_1 H_2 H_3$ color space. The first one cuts clusters orthogonally to the coordinate axis of greatest variance. The second one cuts clusters orthogonally to their major axis. These percentages are obtained by the formula $\frac{FA-MA}{FA}$, where FA represents the squared error obtained by the fixed axis quantization algorithm and MA the squared error obtained by the major axis quantization algorithm.

		1	2	3	4
Cluster choice	Greatest $\boldsymbol{SE}$		$\star$	$\star$	$\star$
	Greatest Cardinal	$\star$			
Set of axes	Coordinate axes	$\star$			$\star$
	Major axis		$\star$	$\star$	
axis choice	Longer axis	$\star$			
	Greatest Variance		$\star$	$\star$	$\star$
Halfplane choice	Median cut	$\star$			
	Variance minimization		$\star$		
	$\boldsymbol{E}()$ minimization			$\star$	$\star$

(1)	:	Heckbert's algorithm.
(2)	:	Wan and Wong's algorithm.
(3)	:	Wu's algorithm.
(4)	:	Proposed algorithm.

TABLE IV

This table summarized the different strategies used by top-down quantization methods.