A Mixed Method with Effective Color Reduction

María-Luisa Pérez-Delgado

Departamento de Informática y Automática, University of Salamanca, Escuela Politécnica Superior de Zamora, Av. Requejo, 33, C.P. 49022 Zamora, Spain; mlperez@usal.es

Received: 28 September 2020; Accepted: 2 November 2020; Published: 4 November 2020

Abstract: This article presents a color quantization technique that combines two previously proposed approaches: the Binary splitting method and the Iterative ant-tree for color quantization method. The resulting algorithm can obtain good quality images with low time consumption. In addition, the iterative nature of the proposed method allows the quality of the quantized image to improve as the iterations progress, although it also allows a good initial image to be quickly obtained. The proposed method was compared to 13 other color quantization techniques and the results showed that it could generate better quantized images than most of the techniques assessed. The statistical significance of the improvement obtained using the new method is confirmed by applying a statistical test to the results of all the methods compared.

Keywords: color quantization; splitting method; clustering-based method

1. Introduction

Color quantization is a process that attempts to reduce the colors of an image. Let us consider a color image including n pixels. When the image is represented using the RGB color space, each pixel p_i of this image, with $1 \leq i \leq n$ is represented by three integer values between 0 and 255 that indicate the amount of red, green and blue that define the pixel, $p_i = (R_i, G_i, B_i)$. Therefore, the palette used to represent this image can include 256^3 different colors. The color quantization problem consists of defining a new palette which includes fewer colors than the palette used to represent the original image. Once this palette containing q colors has been defined, it is used to represent the new image, called quantized image. The quantized palette should be defined in such a way that it allows a new image, as similar to the original as possible, to be obtained.

Color quantization is not only a problem in itself, but it is also related to other operations applied to images, such as image segmentation [1–5], content-based image retrieval [6–8], texture analysis [9–11] and image watermarking [12,13]. When the number of different colors of an image is reduced, the size of the file that contains the image is also reduced. This feature not only reduces the storage space required for the image but also accelerates the speed of transmission of the image. Therefore, although current devices can display images with many colors, reducing the colors of an image is still useful for solving other problems. Several practical scientific and industrial applications that involve color quantization are medical image analysis [14–16], nondestructive inspection of food quality [17,18], biometric recognition [19,20] or surveillance systems [21,22].

The color quantization problem is complex, since the selection of the best colors to define the quantized palette is a NP-complete problem [23]. For this reason, several solution methods have been proposed to solve this problem. With respect to the strategy applied for finding a solution, the methods employed can be classified into two groups: splitting methods and clustering-based methods.

The splitting methods operate on the color cube to define the quantized palette. This cube is progressively divided into smaller boxes until q boxes are obtained. At the end of this process each box defines one color in the quantized palette, and the quantized image is generated by representing the

pixels of each box by the color of said box. This category includes methods such as the Median-cut [24], the Variance-based method [25], the Wu's method [26], the Binary splitting [27] and the Octree [28]. These methods differ in relation to the box that is selected to perform the next division as well as in the splitting axis and the splitting point used to divide the selected box.

Clustering-based methods define clusters (groups) of similar pixels. Each cluster defines a color included within the new palette and that color is used to represent all the pixels of the group in the quantized image. Various clustering methods have been used to reduce the colors of an image. The K-means method is probably the most popular clustering method that has been used for this purpose [29–32]. Neural networks have also been applied to this problem. In addition to the Neuquant method [33], other color quantization methods using Neural networks have been presented in [34–36]. Another group of algorithms based on this approach are those that use a swarm of individuals to solve a complex problem. Some methods of this type applied to color quantization are the Particle swarm optimization algorithm [37,38], the Ant-tree for color quantization (ATCQ) method [39], the Iterative ant-tree for color quantization (ITATCQ) method [40], the Artificial bee colony algorithm combined with K-means [41], the Artificial bee colony algorithm combined with ATCQ [42], the Firefly algorithm combined with ATCQ [43] and the Shuffled-frog leaping algorithm [44]. Other swarm-based methods applied to color quantization are summarized in [45].

Some other methods combine the splitting approach with the clustering-based approach, such as the solution proposed in [46], which combines the Binary splitting (BS) method with the ATCQ algorithm. The resulting method, called BS+ATCQ, combines two existing color quantization techniques and generates better images than each of the two methods applied separately. In addition, it also generates better quality images than other well-known color quantization methods. BS and ATCQ define a tree structure to solve the color quantization problem. When BS+ATCQ is applied, the leaves of the binary tree generated by BS are used to define the initial nodes of the tree considered to apply ATCQ. In general, splitting methods are faster, but clustering-based methods can generate better images. BS+ATCQ takes advantage of both features, since it uses BS to quickly generate an initial solution and then applies ATCQ to improve this solution.

This article proposes the introduction of an improved version of the BS+ATCQ algorithm, by replacing the ATCQ algorithm with the ITATCQ algorithm. The ITATCQ method is based on the ATCQ algorithm but generates better quantized images [40]. Therefore, it is expected that the result obtained by combining ITATCQ with BS will generate better images than those obtained using the initial combination of BS and ATCQ.

This introduction describes some interesting color quantization methods, but obviously it cannot discuss all existing solutions. For a more detailed description of color quantization methods and their operations, [47] can be consulted. This article describes other methods that have been applied to reduce the colors of an image, such as vector quantization, genetic algorithms or competitive learning.

The rest of the article is organized as follows. Section 2.1 describes the BS+ATCQ method, along with the two algorithms used to define it. The ITATCQ algorithm is described in Section 2.2. Then, Section 3 describes the method proposed in this article, while the results obtained after applying it to a set of 24 color images are discussed in Section 4. The proposed method is compared to other color quantification techniques. The quality of the images generated by the different methods is not calculated by means of an error measure based on human perception, but by an error measure that considers the images as arrays of points that are compared with each other. The article finishes by outlining the conclusions drawn.

2. Materials and Methods

2.1. Overview of the BS+ATCQ Algorithm

This section begins by describing the two methods used to define the BS+ATCQ algorithm and then describes this algorithm.

2.1.1. The Binary Splitting Method

This method builds a binary tree to reduce the colors of an image [27]. A binary tree is a data structure where each node can include a maximum of two children. The leaves of the tree created by the algorithm define the quantized palette and identify the subset of pixels of the original image that are represented by the same color in the quantized image.

Each node of this tree represents a subset of pixels of the original image and stores the information used to perform the successive operations. The node j represents the subset of pixels C_j . The color of this node is computed as $color_j = m_j / N_j$, where m_j is the sum of the colors of the pixels in C_j (Equation (1)) and N_j is the number of elements of said set (Equation (2)). In addition, there is a 3×3 matrix associated with the node, denoted as $\mathbf{R}_j$, which stores the sum of the product of each pixel $p_i \in C_j$ and its transpose (Equation (3)).

$$m_j = \sum_{p_i \in C_j} p_i \quad (1)$$

$$N_j = |C_j| \quad (2)$$

$$\mathbf{R}_j = \sum_{p_i \in C_j} p_i p_i^T \quad (3)$$

These three elements allow the covariance of C_j to be computed using Equation (4).

$$\mathbf{R}'_j = \mathbf{R}_j - \frac{1}{N_j} m_j m_j^T \quad (4)$$

Algorithm 1 shows the main steps of BS. At the beginning of the operations, the tree only includes the root node, denoted C_1 , which represents the entire set of pixels of the original image. Once the variables $\mathbf{R}_1$, m_1 and N_1 , which are associated with this node, have been computed, the algorithm applies $q - 1$ iterations to define the q colors of the quantized palette.

Each iteration selects a leaf j , splits the set of pixels C_j into two subsets and then creates two children for the node j , which represent each of the two subsets.

To determine the leaf that will be processed during the current iteration, the eigenvalues of the covariance matrix of the current leaves are considered. Based on this information, the algorithm selects the leaf j with the largest eigenvalue, denoted λ_j .

Once the leaf j has been selected, the two children of this node are created and denoted as jA and jB . Then, the set C_j is divided into two subsets, C_{jA} and C_{jB} , with a plane perpendicular to the direction in which the variation of C_j is the greatest and which passes through the mean point. In this case, such plane is perpendicular to the eigenvector E_j corresponding to the principal eigenvalue λ_j of $\mathbf{R}'_j$. Therefore, the subsets that are associated with the nodes jA and jB are defined as follows: $C_{jA} = \{p_i \in C_j : E_j^T p_i \leq E_j^T color_j\}$ and $C_{jB} = \{p_i \in C_j : E_j^T p_i > E_j^T color_j\}$.

To complete the definition of the new nodes, the three values associated with each one are computed. After applying Equations (1)–(3) to calculate the values for node jA , the values for node jB can be easily calculated using Equations (5)–(7).

$$m_{jB} = m_j - m_{jA} \quad (5)$$

$$N_{jB} = N_j - N_{jA} \quad (6)$$

$$\mathbf{R}_{jB} = \mathbf{R}_j - \mathbf{R}_{jA} \quad (7)$$

when the iterations conclude, the tree has q leaves that define the quantized palette. Then, the quantized image is obtained by representing the pixels associated with each leaf using the color of such leaf.

Algorithm 1 BS algorithm.

```

1: Set  $C_1 = \{p_1, \dots, p_N\}$ 
2: Compute  $m_1$ ,  $N_1$  and  $\mathbf{R}_1$  by Equations (1)–(3)
3: for  $i = 1$  to  $q - 1$  do
4:   Find the leaf  $j$  with the largest  $\lambda_j$ 
5:   Create the two children of node  $j$ 
6:   Split the pixels of  $C_j$  into subsets  $C_{jA}$  and  $C_{jB}$ 
7:   Compute  $m_{jA}$ ,  $N_{jA}$  and  $\mathbf{R}_{jA}$  by Equations (1)–(3)
8:   Compute  $m_{jB}$ ,  $N_{jB}$  and  $\mathbf{R}_{jB}$  by Equations (5)–(7)
9: end for

```

2.1.2. The Ant-Tree for Color Quantization Method

ATCQ is a clustering-based method that is used to reduce the colors of an image [39]. This method represents the pixels of the original image by means of ants and subsequently builds a tree using them based on the similarity among the pixels represented by the ants. The ant h_i represents the pixel p_i of the image, with $1 \leq i \leq n$. At the beginning of the algorithm, all the ants are on the root of the tree, denoted a_0 . The operations of the algorithm allow all the ants to connect to the structure, becoming new nodes of the tree. The number of children of a_0 , denoted as q , is equal to 0 before starting the operations and can increase to a limit defined by Q_{max} . These children, denoted $\{S_1, \dots, S_q\}$, define the root nodes of q subtrees of ants. The node S_j defines a color of the quantized palette, computed as $color_j = sum_j / nc_j$, where sum_j is defined by Equation (8) and nc_j is the number of said ants. When the algorithm concludes, the color of the subtree j is used to represent in the quantized image the pixels of the original image associated with the ants of this subtree.

$$sum_j = \sum_{\substack{h_i \text{ connected} \\ \text{to subtree } j}} p_i \quad (8)$$

The ATCQ operations define an iterative process to connect all the ants to the tree (Algorithm 2). Each iteration takes an ant h_i that is not connected to the structure and operates according to the current position of this ant, denoted a_{pos} .

Algorithm 2 ATCQ algorithm.

```

1: while there are moving ants do
2:   Take a moving ant,  $h_i$ , now on  $a_{pos}$ 
3:   if ( $a_{pos} = a_0$ ) then
4:     Support case operations ( $h_i$ )
5:   else
6:     Ant case operations ( $h_i$ )
7:   end if
8: end while
9: Generate the quantized image

```

If the ant is on the node a_0 , the number of children of this node determines the next operation (Algorithm 3). If a_0 has no children ($q = 0$), the first one is created, and the ant is connected to it. In another case, the selected ant can move to an existing subtree or to a new subtree. To determine the destination of the ant, it is necessary to determine the node S_c with the color most similar to h_i . Let d_{ic} denote this similarity ($d_{ic} = Sim(p_i, color_c)$). In order to make a decision, this value is compared to a threshold computed by Equation (9), where α is a parameter in $(0, 1]$ and e_c is the sum of the similarities between each ant h_i connected to the subtree c and the value of $color_c$ when the ant was

included in the subtree (Equation (10)). If $d_{ic} < T_c$, a new child of a_0 must be created to connect h_i to it. If $d_{ic} \geq T_c$, the ant is included in the subtree c . This last operation is also performed when a new child of a_0 should be created, but it is impossible because this node already has Q_{max} children.

$$T_c = \frac{e_c}{nc_c} \alpha \quad (9)$$

$$e_c = \sum_{\substack{h_i \text{ connected} \\ \text{to subtree } j}} d_{ic} \quad (10)$$

Algorithm 3 Support case operations (h_i)-ATCQ.

```

1: if ( $q = 0$ ) then
2:   Create a new child of  $a_0$  and connect  $h_i$ 
3: else
4:   Select the node  $S_c$  with the color most similar to  $h_i$ 
5:   if ( $d_{ic} < T_c$ ) and ( $q < Q_{max}$ ) then
6:     Create a new child of  $a_0$  and connect  $h_i$ 
7:   else
8:     Update the subtree with root in  $S_c$ 
9:   end if
10: end if
  
```

The similarity between two colors x and y used in [39] is computed as a function of the Euclidean distance, $E(x, y)$, between both colors: $Sim(x, y) = 1 / (1 + E(x, y))$.

Before creating a new child of a_0 , q is increased by one. Then, the three variables associated with the new node S_q are initialized: $nc_q = 1$, $e_q = 0$ and $sum_q = p_i$.

When it is necessary to update an existing subtree c , first the ant h_i is placed on S_c and then the values associated with S_c are updated: $nc_c = nc_c + 1$, $e_c = e_c + d_{ic}$ and $sum_c = sum_c + p_i$.

When the selected ant is not on a_0 , the operations to be performed depend on the number of children of a_{pos} (Algorithm 4). There is a limit to the number of children on the nodes of the tree: Q_{max} for the root node and L_{max} for the others. Both values can be different because Q_{max} limits the size of the quantized palette. If a_{pos} has no children, h_i becomes the first one. If a_{pos} has two children and the second one has never been disconnected from a_{pos} , first this child and all its descendants are disconnected from the tree and moved to a_0 ; then, h_i connects as the new second child. In other cases, a child of a_{pos} is selected, denoted a^+ . If a^+ is not sufficiently similar to h_i , ($Sim(p_i, a^+) < T_c$, where c is the subtree that includes the node a^+), and a_{pos} admits at least one more child, h_i connects to a_{pos} ; in other cases, h_i moves on a^+ .

When the iterations of the algorithm conclude, the nodes of the second level of the tree define a quantized palette that includes q colors, $\{color_1, \dots, color_q\}$. Then, the quantized image is generated by replacing the color of all the ants included in the subtree c with the color of the node S_c .

Algorithm 4 Ant case operations (h_i)-ATCQ.

```

1: if no children connected to  $a_{pos}$  then
2:   Connect  $h_i$  to  $a_{pos}$ 
3: else if two ants connected to  $a_{pos}$  and the second one never disconnected then
4:   Move to  $a_0$  the subtree with root in the second child of  $a_{pos}$ 
5:   Connect  $h_i$  to  $a_{pos}$ 
6: else
7:   Select a node connected to  $a_{pos}, a^+$ 
8:   if ( $\text{Sim}(p_i, a^+) < T_c$ ) and ( $a_{pos}$  has less than  $L_{max}$  children) then
9:     Connect  $h_i$  to  $a_{pos}$ 
10:  else
11:    Move  $h_i$  on  $a^+$ 
12:  end if
13: end if

```

2.1.3. The BS+ATCQ Method

The solution described in [46] applies the ATCQ method to the solution obtained by BS (Algorithm 5). BS defines an initial quantized palette and a partition of the pixels of the original image. This information is used to define an initial tree of ants and ATCQ operations improve the quantized palette.

First, BS creates a binary tree with q leaves, denoted $\{l_1, \dots, l_q\}$. The leaf l_k represents the subset C_{l_k} of pixels of the original image and defines a color of the quantized palette, $color_{l_k} = m_{l_k} / N_{l_k}$, where m_{l_k} and N_{l_k} are computed by Equations (1) and (2), respectively.

Before applying ATCQ, the leaves of the binary tree are used to define q nodes in the second level of the tree of ants, $\{S_1, \dots, S_q\}$. The leaf l_k is used to initialize the three variables associated with the node S_k : nc_k and sum_k take the value of N_{l_k} and m_{l_k} , respectively, while e_k is computed by Equation (11), which computes the similarity between the color of the leaf l_k and each pixel p_i in the subset C_{l_k} .

$$e_k = \sum_{p_i \in C_{l_k}} \text{Sim}(p_i, color_{l_k}) \quad (11)$$

At this point, the tree of ants includes the root node and the maximum number of children allowed for that node ($q = Q_{max}$). Therefore, this tree represents an initial quantized palette with q colors. Then, the ATCQ operations are applied in order to connect all the ants to the tree. When these operations conclude, the quantized image is generated in the same way described for ATCQ.

Algorithm 5 Binary splitting + ATCQ algorithm.

```

1: Apply BS
2: for each leaf  $l_k$  of the binary tree do
3:   Create node  $S_k$  in the second level of the ATCQ tree
4:   Set  $nc_k = N_{l_k}, sum_k = m_{l_k}$ 
5:   Compute  $e_k$  by Equation (11)
6: end for
7: Apply ATCQ
8: Generate the quantized image

```

The computational results reported in [46] show that the image generated by BS+ATCQ is always better than the result of BS or ATCQ applied independently. In addition, this method consumes less time than other methods applied to the same problem.

2.2. The ITATCQ Method

The ITATCQ algorithm is based on the ATCQ algorithm [40]. This method performs the operations of ATCQ during a predefined number of iterations IT (Algorithm 6).

Initially, ATCQ is applied and the tree of ants is defined. After this, all the ants are disconnected from the tree and moved back to the support. Then, the operations of ATCQ are applied again to reconnect the ants to the structure. This process is applied to a certain number of iterations and allows a palette to be defined that is updated as the iterations progress. As a result, the quality of the quantized image obtained increases as the iterations progress.

When the iterative process ends, the quantized image and the quantized palette are defined in the same way described for the ATCQ algorithm.

When ATCQ was proposed, a variant was defined that eliminates the disconnection process in order to accelerate the method [39]. In this case, the method can be further accelerated by simplifying the structure of the tree by including all the ants in the third level. When a 3-level tree is used, the operations of Algorithm 4 reduce to line 2. The computational results reported in [39] showed that the images obtained by this variant are slightly worse than those generated when a multilevel tree is created.

ITATCQ applies the variant of ATCQ that builds a 3-level tree. On the other hand, ITATCQ does not use pixels sorted by average similarity, as was proposed when ATCQ was defined [39]. In this case, the pixels of the image are processed as they are read from the original image (from left to right and from top to bottom).

The results reported in [40] show that ITATCQ generates better images than ATCQ.

Algorithm 6 ITATCQ algorithm.

```

1: for  $i = 1$  to  $IT$  do
2:   while there are moving ants do
3:     Take a moving ant,  $h_i$ , now on  $a_{pos}$ 
4:     if ( $a_{pos} = a_0$ ) then
5:       Support case operations ( $h_i$ )
6:     else
7:       Connect  $h_i$  to  $a_{pos}$ 
8:     end if
9:   end while
10:  if  $t < IT$  then
11:    Move all the ants back to the node  $a_0$ 
12:  end if
13: end for

```

3. The BS+ITATCQ Method

This article proposes a modification to the BS+ATCQ method that consists of replacing ATCQ by ITATCQ. As a result, the new method can obtain a quantized image that improves as the iterations progress. In this way, the user can decide which feature of the algorithm is more important: the quality of the solution or the speed in which a result is obtained.

Therefore, the only step that is modified with respect to BS+ATCQ is the one that applies ATCQ (line 7 of Algorithm 5).

Although ITATCQ generates a 3-level tree to accelerate the process, the creation of a multilevel tree to define BS+ITATCQ has also been considered. Certainly, the results published in [40] show that ITATCQ generates better images if a multilevel tree is considered. This possibility is easy to

implement, since it only requires replacing the operation in line 7 of Algorithm 6 with the operations corresponding to Algorithm 4.

It should be noted that when the original ITATCQ that creates a 3-level tree is combined with BS, the resulting method only depends on the Q_{max} parameter. On the contrary, if the version of ITATCQ that generates a multilevel tree is considered, the resulting method also depends on the parameters α and L_{max} .

Obviously, BS+ITATCQ will consume more time when a multilevel tree is considered. The objective of considering this tree is to analyze whether the improvement in the quality of the image compensates for a longer execution time.

There are many color quantization techniques that combine other existing methods [29,31,32,34,35,37,41–43,46]. Two main characteristics are considered when defining a color quantization method: the quality of the quantized image and the time required to obtain such image. However, it is difficult to achieve both objectives. When several methods are combined to define a new color quantization technique, the selected methods are intended to achieve these objectives or at least one of both.

The solution proposed in this article combines a fast method with another method that can generate a quantized image that improves with the iterations. In this way, the resulting method can generate an image quickly, but such an image can be improved if the method is allowed to consume a little more time. Computational results show that the new method can obtain a quantized image of better quality than BS even when few ITATCQ iterations are executed. Therefore, a small increase in the execution time is required to obtain a better quality image than that generated by BS. On the other hand, if the quality of the final image is more important than the execution time, this method can also achieve this goal, since the quality of the final image improves with the number of iterations of ITATCQ. Computational results also show that the image generated by the new method is better than that generated by ITATCQ applied independently.

Although BS+ATCQ can generate good quantized images quickly, BS+ITATCQ can generate even better images. In addition, the second method allows choosing between speed and quality of the result.

4. Results and Discussion

This section includes results corresponding to tests performed with the proposed method. It also includes the results of other methods described in the literature, in order to compare them to those obtained using the method proposed here. The execution time and the mean squared error (MSE) of the methods are compared. MSE is computed by Equation (12), where p_i represents a pixel of the original image and p'_i represents the pixel in the same position as p_i but in the quantized image.

$$MSE = \frac{1}{n} \sum_{i=1}^n ||p_i - p'_i||^2 \quad (12)$$

The test set includes 24 images used in several research articles, which can be downloaded from [48–51]. Figure 1 shows these images.

Four palette sizes were considered to reduce the colors of the images: {32, 64, 128, 256}. In addition, the six α values proposed in [39] were used to apply ITATCQ: $\alpha = \{0.25, 0.30, 0.35, 0.40, 0.45, 0.50\}$. The tests were executed on a PC running under the Linux operating system with 8 GBytes of RAM and AMD Ryzen 5 1600 processor (3.2 GHz).

The BS+ITATCQ method was executed during 25 iterations. Table A1 shows the results for the variant that considers a 3-level tree (labeled BS+ITATCQ3) while Table A2 includes the results for the variant that considers a multilevel tree (labeled BS+ITATCQm). In order to evaluate the quality of this method compared to other faster methods, both tables show the results at iterations 5 and 25. The results of the tables correspond to 20 independent tests executed for each image, palette size and α value.

Table A2 clearly shows that the error of the image obtained for a given palette size is very similar for all the executions. It is also observed that the average error of BS+ITATCQm is smaller than the error obtained by BS+ITATCQ3 for each image and palette size in almost all the cases. The reduction in the error obtained by the first variant at iteration 25 ranges in $[0.01\%, 5.87\%]$ and only 8 cases exceed 2%. On the other hand, the execution time is longer for BS+ITATCQm, as can be clearly seen in Figure 2, which compares the average execution time for each palette size.

Figure 1. Images used for the tests.

Figure 2. Average execution time (milliseconds) of both variants of BS+ITATCQ at iterations 5 and 25 (q : palette size).

Therefore, BS+ITATCQm can generate better images than BS+ITATCQ3. Nevertheless, since the second variant consumes less time, the rest of the section will consider the results of BS+ITATCQ3 to compare the proposed method to other color quantization methods.

4.1. Results of BS and ITATCQ Applied Independently

The results of BS+ITATCQ3 were compared to those of BS (Table A3) and ITATCQ (Table A4) applied independently. ITATCQ was executed during 25 iterations and the same α values used to test BS+ITATCQ were considered.

It can be clearly observed that the quantized image is of a better quality for the combined method in all cases. Obviously, the combined method consumes more time than each method applied independently.

The amount of error is better for BS+ITATCQ3 than for BS from iteration 5 for all images and palette sizes. The percentage of error reduction obtained by the new method ranges in [9.9%, 30.1%] at iteration 5 and it ranges in [10.9%, 32.3%] at iteration 25. The largest differences obtained when comparing both methods correspond to the Caps image.

The average MSE obtained by ITATCQ at iteration 25 is also worse than the result obtained using BS+ITATCQ3 at iteration 5 for all images and palette sizes. In addition, when comparing the results of iteration 25, the percentage of error reduction obtained by the second method varies in between 17.8% and 93.9%. In this case, the most significant differences correspond to the Mandrill image. Table A4 shows that the average error and the standard deviation of ITATCQ are very large for several images (especially for the Mandrill image). This is because the smallest α values considered do not generate palettes with Q_{max} colors. The most extreme case occurs for the Mandrill image when $\alpha = 0.25$, in which the quantized image only includes one color and the associated error is very large.

The study proposing the ITATCQ method reported that better results are obtained if the range of α values considered for the tests is adjusted according to the size of the quantized palette [40]. Since this was not applied here, a second comparison was performed that considers the minimum MSE obtained by ITATCQ. In this case, the results of BS+ITATCQ3 at iteration 5 are better than those of ITATCQ at iteration 25 in all cases except for 1, corresponding to the Snowman image with 64 colors. In this case the difference is less than 0.3%. Therefore, ITATCQ errors are worse than those of the new method, even if the comparison considers the minimum error obtained by ITATCQ for each image and palette size.

The execution time of BS+ITATCQ3 is not the sum of the execution time of BS and ITATCQ applied independently, as might be expected. This is because the results reported for ITATCQ are the averages for the set of α values tested. The smallest α values generate palettes with less than Q_{max} colors in some cases, so the execution time is reduced. Conversely, when ITATCQ is applied to the result of BS, the initial palette always includes the maximum number of colors allowed.

4.2. Results of BS+ATCQ

The next comparison performed considers the results of BS+ITATCQ3 and BS+ATCQ. The results included in Table A5 correspond to the execution of BS+ATCQ when a 3-level tree is considered, since this case is faster.

The image obtained using the new method is always better. When considering the results of BS+ITATCQ3 at iteration 5, the percentage of error reduction obtained by this method ranges in [0.7%, 27.6%]. On the other hand, this percentage ranges in [2.4%, 28.8%] when the results of iteration 25 are compared.

4.3. Comparison of Results

To complete the previous analysis, Figure 3 shows the average percentage of error reduction obtained by the new method at iteration 25 compared to BS, ITATCQ and BS+ATCQ. In this case, the average percentage is shown for the results grouped by palette size. This figure also shows two results for ITATCQ, labeled ITATCQa and ITATCQm, which use the average and the minimum MSE, respectively. It is clearly observed that all the methods are improved when using BS+ITATCQ. The smallest improvement is obtained when BS+ATCQ is compared. On the other hand, the largest

improvement is obtained when BS+ITATCQ is compared to the average error of ITATCQ. It is also observed that the difference with respect to ITATCQ is reduced by comparing the error of the best image found by ITATCQ instead of the average error.

Figure 3. Average percentage of error reduction of BS+ITATCQ compared to BS, ITATCQ and BS+ATCQ. ITATCQa compares average errors and ITATCQm compares minimum errors (q : palette size).

4.4. Results of Other Methods

Tables A6–A9 show the MSE values obtained using other quantization methods. Four splitting methods have been applied: Wu’s method (WU) [26], Octree (OC) [28], Median-cut (MC) [24] and the Variance-based method (VB) [25]. In addition, five clustering-based methods were also used: Neuquant (NQ) [33], K-means (KM), the Shuffled-frog leaping algorithm (SFLA) [44], the Artificial bee colony algorithm combined with ATCQ (ABC+ATCQ) [42] and the Firefly algorithm combined with ATCQ (ATCQ+FA) [43]. To complete the comparison, the WATCQ method described in [52] was considered, which applies ATCQ to the partition defined by the Wu’s method. The results reported for each method correspond to 20 independent tests for each image and palette size. The results of the iterative methods (KM, ABC+ATCQ, ATCQ+FA and SFLA) correspond to iteration 25 and include the average MSE and the standard deviation. The best solution for each image appears in bold and the worst solution is underlined. To reduce the length of the article, the execution time of each method for each image and palette size is included as Supplementary Materials. However, the article does include a figure that allows the execution time of the methods to be compared (Figure 4). This figure represents the average execution time of each method for each palette size. Two subfigures have been included to show more clearly the time consumed by the fastest methods.

The parameters used for the tests are the following: six food sources have been considered for ABC+ATCQ and six individuals (frogs, fireflies) for SFLA and ATCQ+FA. SFLA was executed considering 2 memplexes, 4 improvement iterations of each memplex and a sampling step equal to 10. The ATCQ algorithm combined with other algorithms was executed considering the variant that generates a 3-level tree, since it is faster. In addition, the α parameter of ATCQ was set to the same values previously described to execute ITATCQ. Finally, the sampling factor of NQ was set to 1, since this is the value that generates the best results for this method.

It should be noted that the results obtained using the ATCQ method are not included because the tests published in [46] have already shown that the BS+ATCQ method improves the images obtained by ATCQ.

The results included in the tables indicate that BS+ITATCQ3 at iteration 5 obtains better MSE than WU, OC, MC, VB and NQ for all the images and palette sizes. In addition, Figure 5 shows the average percentage of error reduction obtained by BS+ITATCQ3 at iterations 5 and 25 compared to those methods. The summary that shows this figure confirms the superiority of BS+ITATCQ3.

Figure 4. Average execution time (milliseconds) of several color quantization methods. The results of BS+ITATCQ in subfigure (a) correspond to the iteration 5, while the results in subfigure (b) correspond to iteration 25 (q : palette size).

Figure 5. Average percentage of error reduction of BS+ITATCQ3 compared to WU, OC, VB, MC and NQ (q : palette size).

When the results of BS+ITATCQ3 at iteration 25 are compared to those of the other iterative methods, the proposed solution obtains worse error in some cases. The best result of the comparison is obtained when ATCQ+FA is considered, since the proposed method obtains better error in all the cases. On the other hand, KM, SFLA and ABC+ATCQ are improved in 85, 77 and 31 cases out of 96, respectively. The percentage of error reduction obtained by BS+ITATCQ3 compared to these methods ranges in [1.15%, 51.29%], [0.01%, 42.56%] and [0.04%, 20.87%], respectively. On the contrary, when these three methods obtain better results than the proposed solution, the percentage of error reduction ranges in [0.11%, 5.49%] for KM, [0.08%, 6.23%] for SFLA and [0.13%, 9.07%] for ABC+ATCQ. It can be clearly observed that the upper limit of the intervals is always considerably greater for the cases improved by BS+ITATCQ3.

When the results of BS+ITATCQ3 at iteration 25 are compared to those of WATCQ, the first method obtains better results for 89 out of 96 cases. These results are similar in the number of cases to those obtained when KM is compared to BS+ITATCQ, although the range is different. In this case the improvement obtained by BS+ITATCQ compared to WATCQ is 10.42% at most.

ABC+ATCQ is the method which obtains better global errors when compared to the proposed method. Nevertheless, it is also the most time consuming method among those considered. It should be noted that the improvement obtained by ABC+ATCQ is less than 5% for the majority of the cases (in 59 out of 65 cases improved by this method).

The results reported for KM correspond to tests that take random initial centroids. However, a second group of tests was performed using the quantized palette generated by BS to define the initial centroids of K-means. The results of these tests appear in Tables A6–A9 labeled as BSKM. Although BSKM and BS+ITATCQ3 have some similarities, there are several differences between both solutions:

- Each iteration of K-means considers an initial set of centroids and uses these centroids to associate each pixel with a cluster. The values of the centroids are the same until the end of the iteration. On the other hand, the colors used by ITATCQ to associate an ant to a subtree are updated during each iteration of the algorithm. When a new ant is associated with a subtree, the color of that subtree is updated to include the color of the ant. Therefore, when the next ant is processed, the set of colors used to decide its destination is different from the set considered for the previous ant.
- K-means recomputes all the centroids at the end of each iteration. In contrast, ITATCQ does not recompute the colors of the palette at the end of each iteration, which accelerates the method.
- The final palette defined by K-means includes the average color of each cluster defined during the last iteration. Nevertheless, the colors of the final palette defined by ITATCQ are computed based on the color of all the ants connected to each subtree throughout all the iterations of the algorithm. When ITATCQ is applied to the result of BS, the colors of the palette defined by BS are also used to compute the final color of each element of the palette.

It can be observed that BSKM obtains better error values than BS+ITATCQ3 in many cases, although it requires much more time. BS+ITATCQ3 obtains better error than BSKM only in 29 out of the 96 cases. For the remaining 67 cases, the reduction in the error obtained by BSKM is less than 4% for 56 cases, ranging between [4%, 6%] for six cases and exceeding 7% for only two cases.

Figure 6 shows the average percentage of error reduction obtained by BS+ITATCQ3 when compared to WATCQ, ATCQ+FA, SFLA, KM, BSKM and ABC+ATCQ. The figure confirms that ABC+ATCQ and BSKM can obtain images with less error than BS+ITATCQ3, although the average difference is less than 4% for all the palettes.

Figure 6. Average percentage of error reduction of BS+ITATCQ3 compared to WATCQ, ATCQ+FA, SFLA, KM, BSKM and ABC+ATCQ (q : palette size).

Figure 7 compares the average MSE obtained by each method for the entire test set. The results are represented considering each palette independently and also considering the results for all the palettes as a whole. The methods were ordered by decreasing error for the case that considers all the palettes as a whole and this order was used in the five subfigures. The results considered for the proposed method are those of BS+ITATCQ3 at iteration 5. This figure confirms that the proposed method is among those that generate the best quality images of all the compared methods.

Figure 7. Average MSE obtained by each quantization method for all the images in the test set.

To complete the comparison, the execution time of the methods must be taken into consideration. The information shown in Figure 4 allows for this comparison. Figure 4a shows the values for the noniterative methods and also includes the results of BS+ITATCQ3 at iteration 5, in order to better interpret the values of this figure. On the other hand, Figure 4b shows the results of the iterative methods. In this second case, the results of BS+ITATCQ3 are those corresponding to iteration 25. To complete the information, Figure 4 also shows the average execution time of BS+ATCQ, BS and ITATCQ, discussed in the previous subsections.

It can be clearly seen that the methods represented in Figure 4a are faster than the method described in this article, although they always generate worse images than the proposed method, except WATCQ. As far as WATCQ is concerned, it only generates better quality images than BS+ITATCQ3 at iteration 5 for 15 cases out of 96, and this value reduces to 7 cases when comparing the results of BS+ITATCQ3 at iteration 25.

Regarding the methods represented in Figure 4b, ABC+ATCQ, ATCQ+FA, KM and BSKM clearly consume more time than BS+ITATCQ3. On the other hand, SFLA is slower than BS+ITATCQ3, but the difference is smaller.

The two desirable features of a color quantization method are its speed and the good quality of the images it generates. To try to compare both features at the same time, a scoring method was defined. First, the average MSE and execution time of each method were computed, considering all the images of the test set and all the palettes. Then, these average values were sorted from worst to best. After this, score 1 was assigned to the method with the worst value and increasingly integer scores were assigned to the sorted methods. As a result, the methods were assigned a score between 1 and 15 when considering the average MSE, while this range was reduced to 1–14 when considering the average execution time, since the execution time is the same for MC and WU methods. Once these two scores were defined, the average of both values was computed. Figure 8 shows the scores assigned for both criteria and the average value. The methods are represented and sorted by increasing average value. It can be clearly observed that BSKM and ABC+ATCQ obtain the best scores when MSE is considered, but they obtain the worst scores when the execution time is considered. On the contrary, MC obtains the best score for the execution time but the worst for MSE. When the average of both scores is considered, WATCQ obtains the best result and the proposed method obtains the second best result.

Figure 8. Scores assigned to the quantization methods, based on the average mean squared error (MSE) and execution time, along with the mean of both scores.

To sum up the results previously presented, the WU, OC, NQ, VB and MC methods are faster than BS+ITATCQ3, but they also generate worse images than BS+ITATCQ3. The WATCQ method is also faster than BS+ITATCQ3, but it generates worse images in most cases. On the contrary, the iterative

methods are slower than BS+ITATCQ3, but some of them can obtain better images than it. Among these methods, ABC+ATCQ and BSKM are the ones that obtain better quality images compared to the proposed method, but they are also the slowest. For example, 25 iterations of BS+ITATCQ3 consume 15 s on average to generate an image with 256 colors. For this same case, BSKM and ABC+ATCQ consume 74 and 162 s, respectively. Therefore, although both methods can obtain better images in many cases, a lot of time is required to obtain a small reduction in the error of the quantized image. However, more iterations of BS+ITATCQ3 could be executed, generating a better quality image and in less time than both methods.

4.5. Statistical Significance of the Results

To conclude the comparison of the methods, the Wilcoxon test [53] was applied in order to determine the statistical significance of the improvement obtained by the proposed method. This test compares the results of two methods to determine that there is no significant difference between both methods.

To compare BS+ITATCQ3 to the iterative methods, the results of iteration 25 of all the methods were used. Nevertheless, when the noniterative methods were compared, the results of BS+ITATCQ3 used were those corresponding to iteration 5. The Wilcoxon test, with a significance level of 0.05, was applied to compare both the MSE values and the execution time. The results of the tests appear in Table 1. The *p*-value (probability value corresponding to the *Z*-value) is not included in the table because it is the same in all cases (<0.00001). This value indicates that the differences between each pair of methods compared are significant in all cases. In addition, the sums of the ranks included in the table allow us to determine which of the two methods compared is better. If the results corresponding to the MSE values are analyzed, it is observed that the proposed method is significantly better than all the other methods, except ABC+ATCQ and BSKM. On the other hand, when analyzing the results for the execution time, the noniterative methods are significantly faster than BS+ITATCQ3. As far as the iterative methods are concerned, the proposed method is significantly faster than all of them, except ITATCQ.

In summary, the results obtained from the Wilcoxon test confirm the conclusions previously presented in this section.

Table 1. Results of the Wilcoxon test that compares BS+ITATCQ3 to other methods. (*Z*-value: value of the test statistic; *sum*+: sum of positive ranks; *sum*−: sum of negative ranks).

Method	MSE			Execution Time		
	Z-Value	sum+	sum−	Z-Value	sum+	sum−
WU	−8.5072	0	4656	−8.5072	4656	0
OC	−8.5072	0	4656	−8.5072	4656	0
VB	−8.5072	0	4656	−8.5072	4656	0
MC	−8.5072	0	4656	−8.5072	4656	0
NQ	−8.5072	0	4656	−8.2496	4585.5	70.5
BS	−8.5072	0	4656	−8.5072	4656	0
WATCQ	−5.4486	837	3819	−8.5072	4656	0
BS+ATCQ	−8.5072	0	4656	−8.5072	4656	0
ITATCQ	−8.5072	0	4656	−8.5072	4656	0
SFLA	−5.3172	847.5	3712.5	−7.7252	214	4442
ATCQ+FA	−8.5072	0	4656	−8.5072	0	4656
KM	−7.7946	195	4461	−8.5072	0	4656
BSKM	−6.1648	4015	641	−8.5072	0	4656
ABC+ATCQ	−4.7122	3549.5	1010.5	−8.5072	0	4656

5. Conclusions

BS+ATCQ is a recently proposed color quantization method which combines two previous methods that solve the same problem: BS and ATCQ. BS is a well-known color quantization technique. Moreover, ATCQ is a recent method that represents the pixels of the image by ants that build a tree structure.

This article proposes to replace ATCQ by ITATCQ, generating another color quantization method called BS+ITATCQ. In essence, ITATCQ applies ATCQ operations iteratively, improving the resulting image as the iterations progress. When ITATCQ was proposed, it was shown that the said algorithm could obtain better images than ATCQ. However, now this work has shown that the method that combines BS with ITATCQ obtains better results than BS+ATCQ.

Computational results show that the new method generates better images than BS and ITATCQ applied independently and also improves the images generated by BS+ATCQ. In addition, the method was compared to 11 other color quantization techniques and obtained good results. BS+ITATCQ obtains better images than the splitting methods compared, although it consumes more time, as is usually the case when comparing the execution time of splitting methods and clustering-based methods. On the other hand, it can obtain better images than most of the clustering-based methods analyzed and with less time consumption.

Supplementary Materials: The following are available online at <http://www.mdpi.com/2076-3417/10/21/7819/s1>, Table S1: Average execution time (milliseconds) of other color quantization methods—palette size 32, Table S2: Average execution time (milliseconds) of other color quantization methods—palette size 64, Table S3: Average execution time (milliseconds) of other color quantization methods—palette size 128, Table S4: Average execution time (milliseconds) of other color quantization methods—palette size 256.

Funding: This research was funded by the Samuel Solórzano Barruso Memorial Foundation of the University of Salamanca grant number FS/102015.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABC	Artificial bee colony
ATCQ	Ant-tree for color quantization
BS	Binary splitting
BSKM	K-means applied to the result of Binary splitting
FA	Firefly algorithm
ITATCQ	Iterative ant-tree for color quantization
KM	K-means
MC	Median-cut
MSE	Mean squared error
NQ	Neuquant
OC	Ocree
SFLA	Shuffled-frog leaping algorithm
VB	Variance-based
WATCQ	Wu's method combined with ant-tree for color quantization
WU	Wu's method

Appendix A

This appendix includes the tables with the main results of the article.

Table A1. MSE and execution time (milliseconds) of BS+ITATCQ3 at iterations 5 and 25 (*AVE.*: average values for all the images).

	Iteration 5								Iteration 25							
	32 Colors		64 Colors		128 Colors		256 Colors		32 Colors		64 Colors		128 Colors		256 Colors	
	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>
Lenna	121.30	549	74.17	927	47.22	1594	30.98	2885	119.98	2051	73.46	3769	46.74	6837	30.63	12,993
Pepp.	245.08	533	147.94	909	94.38	1594	57.96	2873	240.28	1999	144.95	3688	92.12	6889	57.07	12,863
Plane	69.43	485	39.45	867	24.72	1554	15.73	2819	68.52	1922	38.77	3582	24.17	6814	15.47	12,813
Mand.	390.36	586	244.91	935	157.83	1624	101.21	2948	385.83	2131	242.81	3729	156.06	6885	99.94	13,042
Blond	92.15	559	50.94	927	31.26	1603	18.98	2886	90.19	2063	50.15	3759	30.70	6908	18.68	12,962
Lake	212.47	550	137.23	906	88.31	1625	56.62	2888	208.88	2073	134.89	3669	86.84	6896	55.91	12,965
Hydr.	178.15	535	106.23	902	63.25	1585	38.16	3009	174.77	2017	104.79	3626	62.31	6830	37.69	13,211
EPSZ	119.11	551	62.84	917	38.12	1612	23.71	2928	117.59	2068	61.96	3747	37.58	7007	23.43	13,069
Mach.	73.75	538	43.01	894	25.75	1590	15.71	2857	72.44	1983	42.47	3687	25.42	6938	15.55	13,159
Text	35.14	534	20.20	901	12.63	1652	8.15	2946	34.60	2040	19.92	3705	12.53	7095	8.06	13,048
Sign	64.18	525	33.53	881	19.04	1584	11.29	2825	63.36	1992	32.95	3644	18.74	6883	11.15	12,841
Bikes	111.29	544	59.82	900	32.95	1614	18.53	2902	109.37	2061	58.37	3694	32.30	7001	18.24	13,019
Sail.	70.74	781	39.37	1323	20.82	2343	12.04	4236	69.26	2909	38.07	5412	20.32	10,189	11.82	19,138
Moto.	216.31	819	113.86	1364	65.84	2375	39.07	4423	209.61	3039	110.73	5470	64.35	10,182	38.34	19,559
Lady	113.09	806	64.28	1360	35.60	2411	19.98	4393	112.10	2992	63.27	5499	34.82	10,379	19.70	19,591
Caps	168.76	754	78.29	1293	39.60	2317	21.28	4324	161.85	2885	75.85	5370	38.29	10,121	20.79	19,485
Parr.	241.77	791	130.41	1362	77.33	2357	44.27	4298	237.88	2992	128.97	5489	75.53	10,321	43.46	19,362
Girl	135.24	787	74.44	1333	41.74	2383	24.20	4359	133.58	3011	73.33	5430	41.03	10,331	23.80	19,723
Land.	104.87	633	56.53	1054	32.14	1883	19.55	3441	103.12	2394	55.80	4317	31.82	8086	19.36	15,246
Head.	120.85	599	69.36	1042	41.82	1830	24.94	3365	119.15	2326	67.87	4284	41.03	7953	24.52	15,199
Dess.	129.76	605	69.02	1025	40.91	1819	24.15	3306	127.40	2306	68.27	4246	40.42	8049	23.79	15,149
Snow.	130.35	617	65.87	1044	36.66	1860	21.57	3435	128.95	2362	64.79	4245	35.98	8112	21.15	15,218
Cath.	62.12	519	33.20	874	19.08	1561	11.66	2891	60.88	1960	32.59	3576	18.78	6831	11.53	13,235
Beach	146.49	543	76.99	893	44.83	1574	27.43	2974	144.93	2034	75.71	3638	44.35	6853	27.08	13,409
<i>AVE.</i>	139.70	614	78.83	1035	47.16	1831	28.63	3342	137.27	2317	77.53	4220	46.34	7933	28.22	15,012

Table A2. MSE and execution time (milliseconds) of BS+ITATCQm at iterations 5 and 25 (*min*: minimum MSE, *av*: average MSE, *dev*: standard deviation, *it*: iteration of the algorithm, *AVE*.: average values for all the images).

		32 Colors				64 Colors				128 Colors				256 Colors			
it		min	av	dev	T	min	av	dev	T	min	av	dev	T	min	av	dev	T
5	Lenna	120.99	121.07	0.04	1842	74.03	74.07	0.03	2779	47.13	47.16	0.02	4651	30.91	30.92	0.01	8326
	Pepp.	243.04	243.24	0.11	1292	141.19	141.32	0.08	2709	88.25	88.31	0.04	4501	56.66	56.70	0.02	8078
	Plane	66.64	66.69	0.02	1760	38.32	38.36	0.02	2719	23.74	23.75	0.01	4575	15.31	15.32	0.01	8356
	Mand.	388.67	389.03	0.17	1811	244.49	244.61	0.06	2698	157.35	157.48	0.07	4577	100.91	100.99	0.05	8203
	Blond	91.93	92.03	0.06	1812	50.70	50.74	0.02	2694	31.01	31.05	0.02	4669	18.86	18.88	0.01	6313
	Lake	211.61	211.90	0.19	1681	136.54	136.70	0.10	2609	87.99	88.08	0.05	4482	56.04	56.08	0.02	8267
	Hydr.	177.83	178.00	0.08	1737	105.77	105.87	0.05	2740	63.12	63.16	0.03	4543	38.08	38.10	0.01	8221
	EPSZ	118.79	118.85	0.03	1786	62.69	62.73	0.03	2702	38.05	38.09	0.02	4490	23.65	23.66	0.01	8152
	Mach.	73.58	73.63	0.02	1730	42.90	42.94	0.02	2649	25.68	25.70	0.01	4577	15.68	15.68	0.00	8100
	Text	35.02	35.05	0.02	1884	20.13	20.15	0.01	2871	12.61	12.61	0.00	4716	8.12	8.13	0.00	8476
	Sign	64.09	64.13	0.02	1692	33.46	33.49	0.02	2694	18.99	19.00	0.01	4563	11.26	11.27	0.00	8315
	Bikes	110.86	110.98	0.06	1760	59.64	59.72	0.04	2729	32.81	32.87	0.02	4651	18.43	18.46	0.01	8297
	Sail.	70.07	70.18	0.05	2777	37.99	38.05	0.03	4178	20.54	20.58	0.02	7032	11.95	11.96	0.01	12,738
	Moto.	215.11	215.57	0.25	2713	112.89	113.25	0.13	4236	65.44	65.56	0.05	6982	38.72	38.76	0.02	12,336
	Lady	112.84	112.91	0.03	2720	64.08	64.12	0.02	4190	35.43	35.47	0.02	7213	19.90	19.92	0.01	12,791
	Caps	165.25	165.83	0.34	2593	77.52	77.58	0.03	4175	38.75	38.80	0.02	6731	21.00	21.04	0.02	12,286
	Parr.	240.57	240.82	0.14	2686	130.02	130.09	0.05	4062	76.48	76.54	0.03	6852	43.71	43.76	0.02	12,230
	Girl	135.25	135.34	0.03	2776	74.15	74.26	0.05	3998	41.60	41.64	0.02	6864	24.11	24.13	0.01	12,428
	Land.	104.49	104.62	0.07	2190	56.17	56.35	0.06	3278	32.06	32.08	0.01	6214	19.51	19.52	0.01	10,082
	Head.	120.48	120.58	0.07	2119	69.11	69.22	0.05	3534	41.68	41.71	0.03	5678	24.84	24.86	0.01	9768
Dess.	129.01	129.24	0.13	2081	68.90	68.95	0.03	3148	40.79	40.81	0.02	5165	24.08	24.09	0.01	9438	
Snow.	130.06	130.17	0.05	2050	65.57	65.64	0.03	3172	36.51	36.54	0.01	6137	21.40	21.44	0.02	10,092	
Cath.	61.92	61.97	0.03	1696	33.11	33.19	0.05	2801	19.02	19.04	0.01	4588	11.64	11.65	0.00	8332	
Beach	146.48	146.55	0.03	1809	76.78	76.83	0.03	2757	44.70	44.72	0.01	4608	27.33	27.34	0.01	8370	
	AVE.	138.94	139.10		2042	78.17	78.26		3172	46.66	46.70		5377	28.42	28.44		9500
25	Lenna	119.78	119.82	0.02	8477	73.39	73.43	0.02	12,914	46.69	46.71	0.01	22,032	30.59	30.60	0.01	40,102
	Pepp.	240.01	240.14	0.07	5857	139.29	139.39	0.05	12,543	86.96	87.01	0.03	21,357	55.85	55.89	0.02	38,850
	Plane	66.05	66.07	0.01	8203	37.59	37.65	0.03	12,703	23.47	23.50	0.01	21,835	15.09	15.10	0.00	40,434
	Mand.	385.09	385.28	0.10	8252	242.64	242.71	0.03	12,555	155.58	155.69	0.05	21,573	99.72	99.77	0.03	39,536
	Blond	90.11	90.25	0.08	8349	50.10	50.13	0.02	12,539	30.56	30.59	0.02	22,143	18.58	18.59	0.01	30,048
	Lake	208.50	208.61	0.08	7694	134.70	134.78	0.04	12,088	86.60	86.68	0.03	21,349	55.34	55.39	0.02	39,485
	Hydr.	174.54	174.63	0.05	7911	104.51	104.59	0.03	12,774	62.24	62.27	0.03	21,496	37.64	37.66	0.01	39,517
	EPSZ	117.29	117.33	0.02	8122	61.90	61.93	0.03	12,514	37.53	37.55	0.01	21,158	23.38	23.39	0.00	39,113
	Mach.	72.36	72.38	0.01	7897	42.42	42.44	0.01	12,395	25.37	25.38	0.01	21,708	15.53	15.53	0.00	39,164
	Text	34.53	34.56	0.02	8672	19.86	19.88	0.01	13,507	12.52	12.52	0.00	22,502	8.04	8.04	0.00	40,686
	Sign	63.39	63.42	0.01	7732	32.92	32.95	0.01	12,626	18.71	18.72	0.01	21,717	11.12	11.13	0.00	40,168
	Bikes	109.10	109.21	0.05	8115	58.35	58.40	0.02	12,848	32.26	32.29	0.01	22,185	18.17	18.18	0.00	40,064
	Sail.	69.09	69.14	0.02	12,793	36.74	36.81	0.04	19,457	20.12	20.15	0.01	33,285	11.78	11.80	0.01	61,339
	Moto.	201.29	207.41	2.76	12,518	109.68	109.93	0.10	19,678	64.03	64.18	0.07	32,776	38.11	38.13	0.01	59,193
	Lady	112.05	112.09	0.03	12,468	63.15	63.18	0.02	19,600	34.73	34.77	0.02	34,047	19.66	19.67	0.01	62,269
	Caps	160.26	160.37	0.05	11,914	75.44	75.56	0.07	19,856	38.01	38.04	0.01	31,596	20.56	20.59	0.02	58,835
	Parr.	237.27	237.43	0.07	12,125	128.65	128.72	0.03	18,811	75.14	75.18	0.02	32,442	43.05	43.08	0.02	58,919
	Girl	134.15	134.35	0.10	12,836	73.12	73.19	0.03	18,652	40.96	40.99	0.01	32,364	23.72	23.74	0.01	59,553
	Land.	102.87	102.96	0.04	10,011	55.25	55.34	0.08	15,287	31.76	31.77	0.00	29,564	19.35	19.36	0.00	48,674
	Head.	118.85	118.92	0.03	9813	67.76	67.87	0.05	16,498	40.95	41.00	0.03	27,162	24.43	24.45	0.01	47,113
Dess.	126.70	126.85	0.08	9828	68.21	68.24	0.02	14,644	40.28	40.31	0.02	24,605	23.73	23.75	0.01	45,409	
Snow.	128.35	128.53	0.10	9314	64.44	64.59	0.06	14,783	35.89	35.92	0.01	29,336	21.00	21.05	0.02	48,568	
Cath.	60.90	60.93	0.02	7760	32.63	32.66	0.02	12,950	18.74	18.76	0.01	21,844	11.52	11.53	0.00	40,269	
Beach	145.11	145.16	0.02	8192	75.64	75.70	0.02	12,880	44.23	44.26	0.01	22,132	27.01	27.02	0.01	40,599	
	AVE.	136.57	136.91		9369	77.02	77.09		14,796	45.97	46.01		25,509	28.04	28.06		45,746

Table A3. MSE and execution time (milliseconds) of BS (*AVE.*: average values for all the images).

	32 Colors		64 Colors		128 Colors		256 Colors	
	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>	<i>MSE</i>	<i>T</i>
Lenna	138.44	173	82.44	219	53.04	276	35.28	355
Peppers	311.66	165	173.75	214	106.85	272	68.72	344
Plane	83.12	130	46.22	186	28.06	238	18.42	320
Mandrill	441.30	198	286.93	236	183.51	296	117.85	377
Blond	111.28	172	63.07	215	38.46	280	23.65	357
Lake	245.65	167	162.89	213	107.00	273	68.36	355
Hydrant	210.09	163	125.54	219	76.30	269	45.79	341
EPSZ	132.19	170	72.67	219	43.70	274	27.18	353
Machine	86.83	175	49.03	207	29.75	258	18.35	315
Text	43.02	162	23.22	208	14.21	257	9.24	347
Sign	73.66	160	39.97	197	22.86	249	13.41	308
Bikes	130.97	167	72.82	206	39.57	258	21.52	342
Sailboats	86.96	245	47.36	282	27.39	387	14.99	503
Motorcycles	276.14	260	151.98	335	83.79	420	49.37	552
Lady	144.81	255	79.32	310	44.57	388	24.55	504
Caps	217.36	217	112.00	275	55.43	354	28.86	447
Parrots	315.58	234	163.83	292	94.54	353	54.36	466
Girl	151.51	238	89.28	293	52.19	367	30.06	488
Landscape	129.18	197	66.61	238	36.48	316	22.18	416
Headbands	149.52	174	84.86	225	51.30	280	31.28	354
Dessert	152.16	182	85.29	225	48.50	272	29.42	364
Snowman	163.63	184	87.54	242	46.97	296	27.48	385
Cathedrals	72.42	157	38.57	196	22.43	241	13.29	308
Beach	179.49	166	93.22	204	52.30	251	31.89	321
<i>AVE.</i>	168.62	188	95.77	236	56.63	297	34.40	384

Table A4. MSE and execution time (milliseconds) of ITATCQ at iteration 25 (*min*: minimum MSE, *av*: average MSE, *dev*: standard deviation, *AVE.*: average values for all the images).

	32 Colors				64 Colors				128 Colors				256 Colors			
	<i>min</i>	<i>av</i>	<i>dev</i>	<i>T</i>	<i>min</i>	<i>av</i>	<i>dev</i>	<i>T</i>	<i>min</i>	<i>av</i>	<i>dev</i>	<i>T</i>	<i>min</i>	<i>av</i>	<i>dev</i>	<i>T</i>
Len.	137.66	157.21	21.4	1430	78.87	94.83	13.3	2294	50.29	63.45	21.9	3791	32.08	49.67	29.5	6225
Pep.	285.75	350.98	84.9	1529	156.80	177.57	17.0	2627	94.68	118.78	32.7	4325	60.77	93.60	48.8	6873
Pla.	97.59	113.06	16.1	1493	45.38	62.34	15.9	2551	26.63	43.17	17.5	3997	17.10	33.28	22.2	6113
Man.	399.65	1885.24	3568.3	1414	259.82	1768.84	3625.4	2333	167.66	1682.64	3667.6	3482	111.07	1638.11	3689.5	6428
Blo.	94.70	127.46	32.5	1559	59.20	71.70	16.9	2641	36.79	44.59	14.2	4331	21.06	36.08	20.2	6656
Lake	221.71	336.26	177.8	1531	152.24	165.55	11.7	2625	95.28	111.46	23.5	4362	62.13	83.26	38.1	7131
Hyd.	205.22	248.33	39.4	1394	115.76	162.45	42.1	2355	72.48	100.95	31.5	3772	41.45	74.20	41.6	7146
EPS.	146.94	183.01	37.4	1352	78.91	111.61	68.5	2186	41.55	82.30	83.0	3582	25.07	69.35	90.0	6446
Mac.	95.69	183.59	114.3	1371	50.70	84.59	51.1	2258	30.35	60.82	61.3	3687	17.25	50.96	66.4	6284
Text	36.94	58.55	34.6	1380	21.43	28.37	8.8	2168	13.10	20.38	10.5	3309	8.31	17.98	12.4	4836
Sign	89.20	119.38	26.6	1362	41.02	57.71	19.6	2270	20.89	34.82	17.1	3602	12.09	27.18	21.2	6222
Bik.	122.71	145.69	25.3	1382	62.12	80.35	16.4	2350	36.60	48.04	15.9	3797	21.63	37.15	22.8	6476
Sai.	74.26	112.89	33.5	2086	41.40	56.02	18.6	3553	23.88	27.50	3.8	6361	13.92	17.54	6.3	11,043
Mot.	222.81	259.69	35.9	2166	126.42	147.26	14.6	3698	77.49	88.72	15.5	6342	46.17	60.67	28.7	10,633
Lady	135.03	166.36	24.5	2109	71.54	95.44	28.4	3650	42.10	50.83	10.6	6234	23.15	32.60	11.8	10,038
Caps	222.44	287.24	78.5	2030	95.39	136.72	39.6	3556	50.56	72.41	31.0	6322	26.57	38.26	13.8	10,973
Par.	268.19	289.43	19.6	2072	153.54	185.32	20.8	3577	90.19	122.50	22.7	6053	51.67	83.35	32.7	9811
Girl	205.54	267.91	54.6	2025	101.25	117.45	10.7	3542	52.56	67.73	10.0	6400	28.63	39.81	9.7	10,790
Lan.	106.92	201.02	127.6	1370	60.34	95.30	47.5	2238	35.79	62.44	50.2	3588	22.11	53.29	55.6	5877
Hea.	131.38	198.37	91.5	1472	81.18	112.65	32.4	2364	46.39	66.90	26.2	3910	26.61	47.25	30.7	6288
Des.	141.04	189.83	39.3	1396	78.02	96.35	12.9	2418	49.84	61.22	15.3	3991	28.82	44.72	24.8	6647
Sno.	130.51	157.43	16.7	1420	65.69	94.55	21.0	2376	40.46	58.59	28.6	3987	23.67	42.45	34.9	6699
Cat.	74.45	148.43	109.6	1380	39.09	85.89	93.8	2291	22.34	30.99	12.2	3819	13.49	23.40	16.3	6244
Bea.	182.97	369.00	193.8	1396	103.02	249.67	200.2	2371	54.45	122.67	110.1	3936	32.52	57.52	29.1	6604
<i>AVE.</i>	159.55	273.18		1588	89.13	180.77		2679	53.01	135.16		4458	31.97	114.65		7437

Table A5. MSE and execution time (milliseconds) of BS+ATCQ (AVE.: average values for all the images).

	32 Colors		64 Colors		128 Colors		256 Colors	
	MSE	T	MSE	T	MSE	T	MSE	T
Lenna	123.82	250	75.35	362	48.10	541	31.59	864
Peppers	272.24	241	157.17	354	99.05	538	59.75	847
Plane	88.32	203	41.63	325	25.90	504	17.17	821
Mandrill	397.18	278	248.82	378	160.71	561	103.18	931
Blond	95.45	250	70.39	359	42.70	547	21.04	863
Lake	218.92	246	140.32	354	90.70	539	57.65	863
Hydrant	183.95	239	108.71	357	64.93	533	38.96	854
EPSZ	120.74	248	64.55	359	39.09	539	24.21	861
Machine	76.81	250	43.96	345	26.28	521	16.04	819
Text	36.25	238	20.65	347	12.84	522	8.29	854
Sign	65.43	235	34.59	333	19.53	508	11.57	812
Bikes	113.94	243	62.53	346	33.92	521	18.94	854
Sailboats	73.86	357	39.67	491	21.58	781	12.37	1251
Motorcycles	224.66	374	118.62	544	68.55	813	40.17	1314
Lady	116.20	368	66.49	518	36.78	794	20.57	1291
Caps	177.84	328	82.13	483	40.95	750	22.18	1198
Parrots	249.72	345	133.34	498	79.40	757	45.27	1224
Girl	136.95	350	77.10	498	43.26	765	24.87	1265
Landscape	107.72	286	57.59	401	32.69	621	19.87	1019
Headbands	124.51	260	71.23	385	43.06	591	25.76	945
Dessert	133.40	267	70.66	384	41.77	588	24.83	957
Snowman	134.07	271	68.77	403	38.09	604	22.33	981
Cathedrals	64.48	231	34.11	334	19.69	507	11.88	821
Beach	150.04	243	79.09	344	45.78	517	28.11	847
AVE.	145.27	275	81.98	396	48.97	603	29.44	973

Table A6. MSE of other color quantization methods for palette size 32. The results of the iterative methods include the average MSE and the standard deviation (in parentheses). The best solution for each image appears in bold and the worst solution is underlined (AVE.: average values for all the images).

	WU	OC	MC	VB	NQ	BSKM	WATCQ	KM	SFLA	ABC+ATCQ	ATCQ+FA
Lenna	158.61	<u>482.03</u>	425.85	203.07	152.13	118.65	131.41	123.48 (4.6)	119.98 (1.1)	119.19 (0.5)	139.13 (1.2)
Pepp.	279.27	777.14	<u>866.55</u>	451.10	283.69	231.82	241.08	239.02 (7.5)	232.13 (2.3)	231.06 (1.3)	290.13 (4.3)
Plane	85.45	342.23	<u>374.70</u>	123.56	123.70	65.84	65.63	140.67 (28.8)	89.52 (7.6)	68.05 (2.4)	101.35 (0.5)
Mand.	468.39	<u>1094.11</u>	984.05	531.03	456.13	376.69	405.65	382.70 (5.2)	377.53 (2.8)	377.14 (1.4)	408.40 (1.7)
Blond	107.55	<u>477.64</u>	374.69	232.43	123.89	83.92	90.68	89.56 (8.4)	84.58 (1.9)	82.16 (0.3)	96.30 (4.1)
Lake	249.81	922.04	<u>1208.87</u>	357.26	274.45	204.42	215.70	214.49 (7.7)	206.89 (2.5)	204.78 (0.7)	216.94 (1.4)
Hydr.	218.21	491.45	<u>975.28</u>	367.05	235.63	170.52	182.38	179.00 (4.3)	173.76 (2.7)	171.22 (0.9)	200.59 (2.2)
EPSZ	149.84	<u>539.37</u>	291.61	182.44	141.71	114.78	120.61	118.95 (2.6)	115.55 (1.9)	115.37 (1.0)	153.18 (1.0)
Mach.	98.40	<u>371.64</u>	185.28	120.22	96.94	72.73	77.37	77.92 (4.0)	74.30 (1.8)	73.96 (1.0)	94.38 (1.1)
Text	45.51	104.38	<u>250.51</u>	60.96	55.06	34.48	35.96	40.64 (3.0)	36.49 (1.3)	35.46 (0.4)	38.62 (0.1)
Sign	82.07	308.53	<u>412.77</u>	113.41	98.36	61.01	64.92	69.23 (7.2)	65.06 (3.0)	62.41 (1.2)	96.38 (0.9)
Bikes	131.08	409.62	<u>580.27</u>	162.72	140.32	104.31	109.89	115.49 (4.4)	107.70 (2.7)	104.68 (1.3)	124.01 (1.4)
Sail.	85.82	<u>477.06</u>	320.63	139.41	118.56	68.19	70.55	88.38 (6.8)	78.67 (5.3)	69.01 (1.8)	76.18 (0.6)
Moto.	268.42	<u>849.84</u>	532.58	406.60	339.71	187.95	216.60	204.08 (8.0)	197.09 (3.4)	190.87 (1.8)	232.83 (1.5)
Lady	148.83	<u>412.51</u>	317.46	417.87	160.77	111.86	124.17	120.33 (5.2)	116.32 (2.6)	114.45 (1.2)	141.49 (1.0)
Caps	213.01	<u>782.08</u>	637.58	352.56	231.86	157.95	167.25	180.56 (10.5)	171.76 (9.5)	158.08 (5.1)	225.76 (1.2)
Parr.	298.98	<u>853.02</u>	862.83	357.24	297.85	236.24	249.61	253.35 (8.1)	243.10 (5.4)	235.00 (2.8)	269.55 (2.1)
Girl	162.03	<u>508.41</u>	334.74	222.09	169.25	124.93	132.37	144.11 (9.3)	132.00 (3.9)	127.86 (1.5)	186.57 (5.7)
Land.	131.31	<u>576.99</u>	419.37	164.35	139.03	98.74	105.55	103.00 (3.3)	100.33 (1.6)	98.42 (0.7)	111.71 (1.4)
Head.	142.63	430.86	<u>1195.51</u>	184.82	188.99	113.88	120.37	139.68 (9.1)	121.69 (6.0)	114.85 (1.2)	138.24 (0.9)
Dess.	160.65	426.95	<u>451.64</u>	191.72	176.26	123.99	130.46	132.13 (8.4)	124.92 (3.4)	123.12 (1.9)	144.34 (1.5)
Snow.	161.12	<u>559.57</u>	366.86	216.35	202.14	121.41	124.82	137.94 (17.0)	126.51 (9.0)	118.63 (2.0)	137.95 (0.7)
Cath.	81.90	<u>316.46</u>	284.29	105.84	93.72	59.24	63.25	66.76 (4.9)	62.93 (1.7)	60.63 (0.8)	77.27 (0.9)
Beach	177.34	446.43	<u>671.30</u>	211.50	178.32	140.87	145.64	148.69 (10.2)	139.23 (2.3)	138.32 (1.7)	185.36 (3.9)
AVE.	171.09	540.02	555.22	244.82	186.60	132.68	141.33	146.26	137.42	133.11	161.94

Table A7. MSE of other color quantization methods for palette size 64. The results of the iterative methods include the average MSE and the standard deviation (in parentheses). The best solution for each image appears in bold and the worst solution is underlined (*AVE.*: average values for all the images).

	WU	OC	MC	VB	NQ	BSKM	WATCQ	KM	SFLA	ABC+ATCQ	ATCQ+FA
Lenna	99.16	212.92	<u>362.61</u>	135.86	85.60	72.34	79.13	75.81 (2.6)	73.84 (0.8)	72.68 (0.4)	80.56 (1.0)
Peppers	165.37	495.51	<u>771.03</u>	304.21	166.37	136.12	142.07	144.44 (4.0)	139.21 (3.0)	135.31 (1.2)	160.67 (1.1)
Plane	51.33	225.98	<u>237.97</u>	80.73	57.57	37.41	40.59	78.30 (22.7)	52.64 (3.0)	42.53 (1.0)	47.69 (0.4)
Mandrill	288.33	576.19	<u>881.29</u>	346.58	272.25	239.58	249.00	240.61 (2.1)	238.68 (1.4)	237.47 (0.4)	259.32 (2.6)
Blond	63.03	163.08	<u>271.55</u>	129.43	66.32	49.34	51.46	52.33 (1.0)	51.15 (0.7)	49.32 (0.4)	58.05 (0.8)
Lake	161.34	466.14	<u>984.62</u>	251.70	164.26	131.72	139.07	137.06 (1.7)	134.11 (1.8)	131.58 (0.6)	154.10 (0.7)
Hydrant	131.49	280.76	<u>801.44</u>	236.15	127.00	102.02	109.45	108.14 (2.8)	104.91 (1.4)	102.12 (0.6)	116.85 (1.0)
EPSZ	85.95	<u>280.23</u>	232.32	113.99	72.76	61.76	66.56	65.84 (2.9)	62.73 (1.0)	61.59 (0.5)	82.94 (1.8)
Machine	57.81	136.94	<u>156.20</u>	82.35	53.35	42.65	44.80	45.67 (1.5)	43.91 (1.0)	42.89 (0.6)	52.10 (1.0)
Text	27.26	72.16	<u>165.16</u>	42.87	28.58	20.48	21.17	24.02 (1.6)	21.58 (0.6)	20.68 (0.4)	22.87 (0.2)
Sign	45.21	121.85	<u>273.21</u>	62.61	52.50	33.15	35.39	37.87 (2.1)	34.65 (1.0)	33.04 (0.3)	42.56 (0.9)
Bikes	77.65	193.87	<u>429.39</u>	103.25	75.52	56.53	59.87	64.60 (2.3)	61.21 (2.0)	57.75 (0.4)	64.30 (0.6)
Sailb.	50.30	144.83	<u>200.83</u>	71.01	59.26	36.57	36.65	49.34 (3.8)	48.90 (3.1)	36.75 (0.7)	43.77 (1.1)
Motor.	147.10	351.29	<u>398.09</u>	216.75	140.49	107.09	115.86	118.23 (5.4)	114.34 (3.0)	108.60 (0.8)	131.05 (1.0)
Lady	82.36	<u>333.21</u>	231.33	315.04	81.23	62.42	64.84	67.02 (3.0)	66.95 (1.7)	61.18 (0.6)	72.15 (1.0)
Caps	102.70	323.01	<u>431.91</u>	163.94	93.91	73.63	79.62	94.48 (8.4)	93.74 (6.9)	77.86 (2.1)	99.69 (1.9)
Parrots	167.24	364.27	<u>609.94</u>	217.62	156.24	128.15	136.67	138.16 (6.1)	133.91 (2.7)	127.66 (1.0)	143.51 (3.0)
Girl	93.38	<u>328.70</u>	258.61	119.52	93.44	71.68	76.69	82.18 (3.8)	78.05 (3.2)	71.99 (0.6)	102.75 (1.1)
Lands.	72.20	<u>185.19</u>	<u>352.09</u>	113.61	70.13	54.56	58.21	58.69 (2.1)	56.43 (1.0)	54.10 (0.3)	62.75 (0.7)
Headb.	87.52	192.74	<u>878.94</u>	121.28	99.49	66.24	70.63	83.25 (4.9)	76.88 (3.6)	68.03 (0.7)	86.48 (0.5)
Dessert	90.42	203.65	<u>311.91</u>	118.69	90.70	67.40	71.30	75.90 (4.0)	70.10 (1.9)	68.45 (0.7)	82.38 (0.9)
Snowm.	89.53	<u>334.53</u>	305.82	118.13	90.05	62.12	66.87	72.57 (3.4)	67.47 (2.3)	62.30 (1.1)	66.96 (0.3)
Cathe.	45.28	109.02	<u>169.33</u>	61.50	48.67	32.89	34.33	36.43 (1.8)	35.72 (1.7)	32.55 (0.4)	41.55 (0.3)
Beach	101.92	309.23	<u>557.04</u>	123.49	91.86	74.08	81.68	79.90 (3.2)	76.28 (1.8)	74.63 (0.6)	99.81 (1.8)
AVE.	99.33	266.89	428.03	152.10	97.40	75.83	80.50	84.62	80.72	76.29	90.62

Table A8. MSE of other color quantization methods for palette size 128. The results of the iterative methods include the average MSE and the standard deviation (in parentheses). The best solution for each image appears in bold and the worst solution is underlined (*AVE.*: average values for all the images).

	WU	OC	MC	VB	NQ	BSKM	WATCQ	KM	SFLA	ABC+ATCQ	ATCQ+FA
Lenna	61.79	140.53	<u>276.33</u>	94.68	53.73	46.80	49.54	48.52 (0.8)	47.83 (0.4)	46.41 (0.1)	50.98 (0.5)
Peppers	102.31	308.76	<u>585.41</u>	212.68	95.69	84.56	88.64	87.06 (2.9)	86.92 (1.8)	83.76 (0.3)	98.31 (0.6)
Plane	32.60	133.59	<u>187.93</u>	52.60	29.71	23.82	25.20	42.00 (11.6)	33.99 (1.6)	28.59 (0.6)	27.90 (0.2)
Mandrill	186.33	357.13	<u>776.65</u>	248.02	168.22	151.75	162.45	154.46 (1.1)	154.53 (0.8)	152.54 (0.2)	167.23 (2.2)
Blond	36.84	89.86	<u>185.81</u>	82.22	38.30	30.32	31.17	33.24 (0.8)	33.23 (0.7)	30.74 (0.7)	37.26 (1.0)
Lake	102.55	198.49	<u>692.56</u>	170.97	100.88	84.77	88.50	90.78 (1.0)	89.54 (0.9)	85.31 (0.4)	100.02 (1.3)
Hydrant	79.91	208.59	<u>643.83</u>	175.19	74.10	61.20	65.92	66.14 (1.7)	65.38 (1.3)	61.07 (0.3)	72.75 (0.5)
EPSZ	52.04	124.29	<u>179.43</u>	78.97	43.26	37.33	40.12	39.10 (0.8)	38.28 (0.5)	36.98 (0.2)	42.74 (0.4)
Machine	35.59	85.40	<u>128.24</u>	60.15	30.63	25.54	27.37	27.79 (0.7)	26.92 (0.4)	25.47 (0.2)	32.05 (0.6)
Text	18.76	40.46	<u>105.83</u>	33.12	15.95	13.24	13.49	15.03 (0.6)	14.11 (0.5)	13.14 (0.1)	14.09 (0.1)
Sign	26.98	57.33	<u>168.87</u>	38.48	26.16	18.86	20.09	22.35 (1.0)	21.26 (0.7)	18.91 (0.2)	22.04 (0.2)
Bikes	44.62	129.45	<u>331.58</u>	65.29	39.23	31.66	33.83	37.13 (1.1)	36.55 (1.2)	32.26 (0.4)	37.91 (0.4)
Sailb.	29.23	118.13	<u>176.83</u>	44.09	28.26	20.20	21.27	28.20 (1.7)	30.23 (1.2)	20.64 (0.3)	25.57 (0.2)
Motor.	86.71	230.56	<u>299.41</u>	119.36	79.60	62.84	68.14	71.87 (2.0)	71.26 (1.8)	62.78 (0.7)	80.70 (0.8)
Lady	46.69	126.93	<u>168.37</u>	189.51	41.66	34.21	36.28	37.78 (1.0)	38.64 (1.0)	33.91 (0.3)	43.05 (0.7)
Caps	51.90	171.52	<u>305.15</u>	75.92	49.04	37.22	40.00	51.30 (3.1)	52.72 (2.4)	39.00 (0.7)	54.15 (0.8)
Parrots	95.39	253.69	<u>441.59</u>	143.31	86.03	72.74	77.44	79.53 (3.5)	78.72 (1.5)	72.78 (0.6)	86.42 (1.3)
Girl	54.95	169.86	<u>200.69</u>	80.14	51.39	40.86	43.62	50.67 (2.4)	49.01 (1.3)	40.46 (0.5)	55.13 (0.6)
Lands.	42.75	149.46	<u>257.61</u>	84.46	37.17	32.06	33.37	35.19 (1.5)	34.56 (0.8)	31.30 (0.3)	37.27 (0.3)
Headb.	53.42	128.22	<u>519.00</u>	79.90	52.12	39.88	43.05	51.82 (2.7)	49.26 (1.8)	40.54 (0.3)	48.75 (0.3)
Dessert	52.66	118.32	<u>233.09</u>	82.38	51.09	39.82	42.51	44.60 (1.7)	43.36 (1.0)	40.24 (0.5)	52.36 (1.0)
Snowm.	49.69	134.31	<u>212.60</u>	70.03	44.23	35.69	37.75	41.95 (1.3)	41.45 (1.3)	35.27 (0.5)	42.25 (0.3)
Cathe.	27.31	69.79	<u>142.12</u>	40.15	24.33	19.21	20.31	22.05 (0.8)	21.29 (0.5)	18.74 (0.2)	23.99 (0.7)
Beach	59.64	134.36	<u>446.84</u>	81.50	52.71	44.15	47.07	48.07 (1.5)	46.82 (1.0)	44.59 (0.6)	55.23 (0.6)
AVE.	59.61	153.29	319.41	100.13	54.73	45.36	48.21	51.11	50.24	45.64	54.51

Table A9. MSE of other color quantization methods for palette size 256. The results of the iterative methods include the average MSE and the standard deviation (in parentheses). The best solution for each image appears in bold and the worst solution is underlined (AVE.: average values for all the images).

	WU	OC	MC	VB	NQ	BSKM	WATCQ	KM	SFLA	ABC+ATCQ	ATCQ+FA
Lenna	39.53	74.12	<u>193.25</u>	69.41	34.88	30.79	32.16	31.96 (0.4)	31.91 (0.3)	30.32 (0.1)	33.39 (0.2)
Peppers	66.08	156.24	<u>397.29</u>	147.18	63.08	54.79	56.90	56.77 (2.2)	57.03 (0.5)	53.95 (0.1)	63.18 (0.7)
Plane	21.66	52.31	<u>147.53</u>	36.85	21.09	15.50	15.93	23.72 (1.2)	23.53 (1.1)	19.55 (0.4)	18.12 (0.1)
Mandrill	118.65	195.82	<u>676.87</u>	181.94	109.34	97.52	102.82	99.08 (0.6)	100.40 (0.6)	97.70 (0.1)	108.13 (0.4)
Blond	23.59	50.79	<u>123.57</u>	54.42	23.86	18.87	19.89	22.08 (0.7)	21.76 (0.9)	19.19 (0.7)	22.11 (0.2)
Lake	66.47	159.21	<u>523.09</u>	120.10	65.70	54.63	57.26	60.38 (1.5)	60.20 (0.9)	54.94 (0.2)	63.48 (0.4)
Hydrant	48.62	120.00	<u>459.48</u>	118.85	45.88	37.51	39.94	41.57 (0.7)	41.81 (0.6)	37.34 (0.2)	43.19 (0.4)
EPSZ	32.00	55.06	<u>134.55</u>	57.37	27.45	23.72	24.96	24.94 (0.4)	24.73 (0.3)	23.37 (0.1)	26.87 (0.4)
Machine	22.02	49.07	<u>96.87</u>	41.62	19.73	16.08	16.67	17.31 (0.2)	17.12 (0.2)	15.65 (0.1)	18.68 (0.2)
Text	14.05	27.43	<u>73.00</u>	26.48	10.06	8.76	9.00	10.01 (0.3)	9.38 (0.2)	8.57 (0.1)	9.01 (0.0)
Sign	17.23	42.01	<u>94.21</u>	25.79	14.43	11.78	12.02	13.70 (0.4)	13.51 (0.3)	11.41 (0.1)	13.23 (0.1)
Bikes	26.12	77.85	<u>243.34</u>	41.30	23.65	18.60	19.34	22.39 (0.5)	23.02 (0.6)	18.73 (0.2)	22.42 (0.7)
Sailb.	18.89	41.05	<u>126.59</u>	27.62	18.00	12.63	13.00	18.06 (0.9)	20.58 (0.8)	12.41 (0.1)	14.94 (0.1)
Motor.	51.14	126.16	<u>228.45</u>	73.01	46.93	37.86	39.81	44.41 (1.2)	46.08 (1.5)	37.18 (0.1)	49.22 (0.5)
Lady	27.34	78.17	<u>121.81</u>	90.16	23.27	19.95	20.64	22.04 (0.4)	24.01 (0.7)	19.00 (0.1)	24.03 (0.4)
Caps	29.68	70.45	<u>212.22</u>	43.02	27.73	20.95	21.77	30.10 (1.4)	32.34 (1.6)	21.52 (0.4)	27.76 (0.7)
Parrots	58.43	123.65	<u>311.14</u>	98.34	50.41	42.55	46.21	47.77 (1.4)	48.17 (1.4)	42.16 (0.2)	53.20 (1.1)
Girl	32.96	109.06	<u>159.37</u>	53.84	30.80	23.85	25.29	30.97 (0.9)	31.36 (1.0)	23.55 (0.2)	30.61 (0.6)
Lands.	25.75	53.17	<u>185.22</u>	61.90	23.75	19.88	20.40	22.47 (0.7)	22.74 (0.4)	19.18 (0.1)	23.24 (0.2)
Headb.	33.83	53.17	<u>314.55</u>	53.29	30.77	24.74	26.17	33.09 (1.8)	32.90 (0.6)	25.75 (0.2)	28.25 (0.1)
Dessert	32.71	67.32	<u>172.45</u>	56.92	30.21	23.92	25.36	27.43 (0.9)	27.09 (0.6)	23.86 (0.2)	30.45 (1.0)
Snowm.	29.85	84.45	<u>165.97</u>	43.68	27.51	21.36	22.34	25.63 (0.6)	25.86 (0.7)	21.15 (0.1)	25.62 (0.3)
Cathe.	18.10	45.83	<u>99.17</u>	26.16	15.28	12.21	12.64	13.98 (0.2)	14.19 (0.4)	11.74 (0.1)	14.71 (0.3)
Beach	36.33	81.77	<u>292.27</u>	52.92	32.95	27.21	28.47	30.22 (0.6)	29.66 (0.7)	27.50 (0.3)	33.63 (0.4)
AVE.	37.13	83.09	231.34	66.76	34.03	28.15	29.54	32.09	32.47	28.16	33.23

References

1. An, N.Y.; Pun, C.M. Color image segmentation using adaptive color quantization and multiresolution texture characterization. *Signal Image Video Process.* **2014**, *8*, 943–954. [\[CrossRef\]](#)
2. Deng, Y.; Manjunath, B. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 800–810. [\[CrossRef\]](#)
3. Nguyen, T.D.; Lee, G. Color image segmentation using tensor voting based color clustering. *Pattern Recognit. Lett.* **2012**, *33*, 605–614. [\[CrossRef\]](#)
4. Phung, S.L.; Bouzerdoun, A.; Chai, D. Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 148–154. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Yu, Z.; Au, O.C.; Zou, R.; Yu, W.; Tian, J. An adaptive unsupervised approach toward pixel clustering and color image segmentation. *Pattern Recognit.* **2010**, *43*, 1889–1906. [\[CrossRef\]](#)
6. Jeong, S.; Won, C.S.; Gray, R.M. Image retrieval using color histograms generated by Gauss mixture vector quantization. *Comput. Vis. Image Underst.* **2004**, *94*, 44–66. [\[CrossRef\]](#)
7. Liu, G.H.; Yang, J.Y. Content-based image retrieval using color difference histogram. *Pattern Recognit.* **2013**, *46*, 188–198. [\[CrossRef\]](#)
8. Singha, M.; Hemachandran, K. Content based image retrieval using color and texture. *Signal Image Process.* **2012**, *3*, 39. [\[CrossRef\]](#)
9. Losson, O.; Macaire, L. CFA local binary patterns for fast illuminant-invariant color texture classification. *J. Real-Time Image Process.* **2015**, *10*, 387–401. [\[CrossRef\]](#)
10. Ponti, M.; Nazaré, T.S.; Thumé, G.S. Image quantization as a dimensionality reduction procedure in color and texture feature extraction. *Neurocomputing* **2016**, *173*, 385–396. [\[CrossRef\]](#)
11. Sertel, O.; Kong, J.; Lozanski, G.; Shana'ah, A.; Catalyurek, U.; Saltz, J.; Gurcan, M. Texture classification using nonlinear color quantization: Application to histopathological image analysis. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008), Las Vegas, NV, USA, 31 March–4 April 2008; pp. 597–600.

12. Kuo, C.T.; Cheng, S.C. Fusion of color edge detection and color quantization for color image watermarking using principal axes analysis. *Pattern Recognit.* **2007**, *40*, 3691–3704. [[CrossRef](#)]
13. Tsai, P.; Hu, Y.C.; Chang, C.C. A color image watermarking scheme based on color quantization. *Signal Process.* **2004**, *84*, 95–106. [[CrossRef](#)]
14. Hance, G.; Umbaugh, S.; Moss, R.; Stoecker, W. Unsupervised color image segmentation: With application to skin tumor borders. *IEEE Eng. Med. Biol. Mag.* **1996**, *15*, 104–111. [[CrossRef](#)]
15. Maitra, I.; Nag, S.; Bandyopadhyay, S. Accurate breast contour detection algorithms in digital mammogram. *Int. J. Comput. Appl.* **2011**, *25*, 1–13. [[CrossRef](#)]
16. Stanley, R.; Moss, R.; Van Stoecker, W.; Aggarwal, C. A fuzzy-based histogram analysis technique for skin lesion discrimination in dermatology clinical images. *Comput. Med. Imaging Graph.* **2003**, *27*, 387–396. [[CrossRef](#)]
17. Lee, D.J.; Archibald, J.; Xiong, G. Rapid color grading for fruit quality evaluation using direct color mapping. *IEEE Trans. Autom. Sci. Eng.* **2010**, *8*, 292–302. [[CrossRef](#)]
18. Lee, D.J.; Chang, Y.; Archibald, J.; Greco, C. Color quantization and image analysis for automated fruit quality evaluation. In Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE 2008), Arlington, VA, USA, 23–26 August 2008; pp. 194–199. [[CrossRef](#)]
19. Abate, A.; Frucci, M.; Galdi, C.; Riccio, D. BIRD: Watershed based iris detection for mobile devices. *Pattern Recognit. Lett.* **2015**, *57*, 43–51. [[CrossRef](#)]
20. D'Angelo, A.; Dugelay, J.L. People re-identification in camera networks based on probabilistic color histograms. In Proceedings of the Visual Information Processing and Communication II. International Society for Optics and Photonics, San Francisco, CA, USA, 23–27 January 2011; Volume 7882, pp. 1–12. [[CrossRef](#)]
21. Cheon, Y.; Lee, C. License plate extraction for moving vehicles. In Proceedings of the 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 15–17 July 2019; pp. 1–6.
22. Moon, H.M.; Pan, S. A new human identification method for intelligent video surveillance system. In Proceedings of the 2010 19th International Conference on Computer Communications and Networks, Zurich, Switzerland, 2–5 August 2010; pp. 1–6.
23. Garey, M.; Johnson, D.; Witsenhausen, H. The complexity of the generalized Lloyd-max problem (corresp). *IEEE Trans. Inf. Theory* **1982**, *28*, 255–256. [[CrossRef](#)]
24. Heckbert, P. Color Image Quantization for Frame Buffer Display. In Proceedings of the 9th Annual Conference on Computer Graphics and Interactive Techniques, Boston, MA, USA, 26–30 July 1982; ACM: New York, NY, USA, 1982; pp. 297–307. [[CrossRef](#)]
25. Wan, S.; Prusinkiewicz, P.; Wong, S. Variance-based color image quantization for frame buffer display. *Color Res. Appl.* **1990**, *15*, 52–58. [[CrossRef](#)]
26. Wu, X. Efficient Statistical Computations for Optimal Color Quantization. In *Graphics Gems II*; Arvo, J., Ed.; Academic Press: San Diego, CA, USA, 1991; pp. 126–133. [[CrossRef](#)]
27. Orchard, M.; Bouman, C. Color quantization of images. *IEEE Trans. Signal Process.* **1991**, *39*, 2677–2690. [[CrossRef](#)]
28. Gervautz, M.; Purgathofer, W. A Simple Method for Color Quantization: Octree Quantization. In *Graphics Gems*; Glassner, A.S., Ed.; Academic Press Professional, Inc.: San Diego, CA, USA, 1990; pp. 287–293. [[CrossRef](#)]
29. Celebi, M. Improving the performance of k-means for color quantization. *Image Vis. Comput.* **2011**, *29*, 260–271. [[CrossRef](#)]
30. Kasuga, H.; Yamamoto, H.; Okamoto, M. Color quantization using the fast K-means algorithm. *Syst. Comput. Jpn.* **2000**, *31*, 33–40. [[CrossRef](#)]
31. Verevka, O.; Buchanan, J. The Local K-Means Algorithm for Colour Image Quantization. In Proceedings of the Graphics Interface 95, Quebec, QC, Canada, 17–19 May 1995; pp. 128–135. [[CrossRef](#)]
32. Wen, Q.; Celebi, M. Hard versus fuzzy c-means clustering for color quantization. *EURASIP J. Adv. Signal Process.* **2011**, *2011*, 118. [[CrossRef](#)]
33. Dekker, A. Kohonen neural networks for optimal colour quantization. *Netw. Comput. Neural Syst.* **1994**, *5*, 351–367. [[CrossRef](#)]

34. Atsalakis, A.; Papamarkos, N. Color reduction and estimation of the number of dominant colors by using a self-growing and self-organized neural gas. *Eng. Appl. Artif. Intell.* **2006**, *19*, 769–786. [CrossRef]
35. Chang, C.H.; Xu, P.; Xiao, R.; Srikanthan, T. New adaptive color quantization method based on self-organizing maps. *IEEE Trans. Neural Netw.* **2005**, *16*, 237–249. [CrossRef] [PubMed]
36. Palomo, E.; Domínguez, E. Hierarchical color quantization based on self-organization. *J. Math. Imaging Vis.* **2014**, *49*, 1–19. [CrossRef]
37. Omran, M.; Engelbrecht, A.; Salman, A. A color image quantization algorithm based on particle swarm optimization. *Informatica (Slovenia)* **2005**, *29*, 261–270.
38. Pérez-Delgado, M.L. Color quantization with Particle swarm optimization and artificial ants. *Soft Comput.* **2020**, *24*, 4545–4573. [CrossRef]
39. Pérez-Delgado, M.L. Colour quantization with Ant-Tree. *Appl. Soft Comput.* **2015**, *36*, 656–669. [CrossRef]
40. Pérez-Delgado, M.L. An iterative method to improve the results of Ant-tree algorithm applied to colour quantisation. *Int. J. Bio-Inspir. Comput.* **2018**, *12*, 87–114. [CrossRef]
41. Ozturk, C.; Hancer, E.; Karaboga, D. Color image quantization: A short review and an application with artificial bee colony algorithm. *Informatica* **2014**, *25*, 485–503. [CrossRef]
42. Pérez-Delgado, M.L. The color quantization problem solved by swarm-based operations. *Appl. Intell.* **2019**, *49*, 2482–2514. [CrossRef]
43. Pérez-Delgado, M.L. Artificial ants and fireflies can perform colour quantisation. *Appl. Soft Comput.* **2018**, *73*, 153–177. [CrossRef]
44. Pérez-Delgado, M.L. Color image quantization using the shuffled-frog leaping algorithm. *Eng. Appl. Artif. Intell.* **2019**, *79*, 142–158. [CrossRef]
45. Pérez-Delgado, M.L. Recent Applications of Swarm-Based Algorithms to Color Quantization. In *Recent Advances on Memetic Algorithms and Its Applications in Image Processing*; Springer: Singapore, 2020; pp. 93–118.
46. Pérez-Delgado, M.L.; Román Gallego, J.Á. A two-stage method to improve the quality of quantized images. *J. Real-Time Image Proc.* **2018**, *17*, 581–605. [CrossRef]
47. Brun, L.; Trémeau, A. *Digital Color Imaging Handbook. Color Quantization*; CRC Press: Boca Raton, FL, USA, 2003; pp. 589–638.
48. Franzen, R. Kodak Lossless True Color Image Suite. 2017. Available online: <http://r0k.us/graphics/kodak/> (accessed on 21 June 2012).
49. Pérez-Delgado, M.L. Images for Color Quantization. 2017. Available online: <http://audax.zam.usal.es/web/mlperez/cq.htm> (accessed on 21 June 2020).
50. Pérez-Delgado, M.L. Campus Viriato Images. 2018. Available online: http://audax.zam.usal.es/web/mlperez/fotos_campus.html (accessed on 21 June 2020).
51. Weber, A. USC-SIPI Image Database. 2020. Available online: <http://sipi.usc.edu/database/database.php/> (accessed on 21 June 2020).
52. Pérez-Delgado, M.L.; Román Gallego, J.Á. A hybrid color quantization algorithm that combines the Greedy orthogonal bi-partitioning method with artificial ants. *IEEE Access* **2019**, *7*, 128714–128734. [CrossRef]
53. Corder, G.; Foreman, D., Comparing two related samples: The Wilcoxon signed ranks test. In *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*; Wiley Online Library: Hoboken, NJ, USA, 2009; pp. 38–56.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).