Color Image Quantization by Pairwise Clustering

LUIZ VELHO[1], JONAS GOMES[1], MARCOS VINICIUS RAYOL SOBREIRO[1,2]

[1] IMPA–Instituto de Matemática Pura e Aplicada, Estrada Dona Castorina, 110, 22460 Rio de Janeiro, RJ
{lvelho,jonas,mrayol}@visgraf.impa.br

[2] PUC-Rio, Pontifícia Universidade Católica, Rio de Janeiro, RJ, Brasil
mrayol@inf.puc-rio.br

Abstract. This paper presents a new quantization method for color images. It uses a local error optimization strategy to generate near optimal quantization levels. The algorithm is simple to implement and produces results that are superior than those of other popular image quantization algorithms.
Keywords: Image Processing; Color Quantization; Otimization; Clustering.

1 Introduction

The quantization of a digital image consists in discretizing the image's color gamut, which implies in a reduction of the color information associated with each pixel in the image.

Color image quantization is an important problem in Computer Graphics and Image Processing. One practical reason for quantization is related to image display. In order to represent an image in a graphics device, the image's color gamut cannot be greater than the gamut of the device. It is very common to find display devices that are capable of displaying only 256 colors. To be displayed in these devices, images must be quantized to 8 bits.

On the other hand, the discretization of color information reduces the image size and is closely related to compression. Quantization in itself does not constitute an effective compression technique, but it is present as an important part of several lossy image compression techniques.

The ultimate goal of quantization is to change the color resolution of an image (number of bits in the color representation) with minimum distortion. This problem involves a variety of theoretical issues, related with color perception and optimization methods. The complexity of the problem makes the computation of an optimal solution is not feasible in general. For this reason, existing quantization methods usually produce only approximate results.

We propose a new color image quantization method that iterates a local optimization technique to reduce the number of colors in the image's gamut, until the desired number of color quantization levels is achieved. The algorithm is simple to implement and produces results that are far superior to other popular image quantization algorithms, such as the median cut [1].

2 Background

A *color quantization* is a map $q\colon \mathbb{R}^n \to C'$, where $\mathbb{R}^n$ is a finite-dimensional color space representation, and $C' \subset \mathbb{R}^n$ is a finite subset of colors of the space. The map q is called a *quantization map*. In practice we need to define a quantization map $q : C \to C'$ from some finite set of colors $C \subset \mathbb{R}^n$, with M colors, onto a set $C' \subset \mathbb{R}^n$ of N colors, with $N < M$.

Consider a quantization map $q\colon C \to C'$. For a given color $c_i \in C'$ the inverse image

$$C_i = q^{-1}(c_i) = \{c \in C : q(c) = c_i\}. \qquad (1)$$

is called a *quantization cell*. The color value c_i is called the *quantization level* of the cell C_i. The quantization map q defines a family of quantization cells that constitutes a partition of the color space C. Figure 1 illustrates a partition of a 2-dimensional color space into quantization cells, with the associated quantization levels.

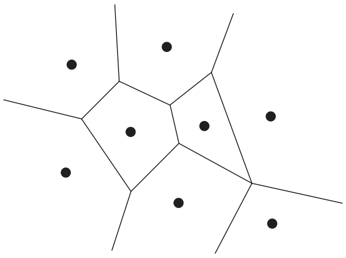

Figure 1: Quantization cells with associated levels.

The quantization of a color space, involves two parts: determining the quantization cells and determining the quantization level for each cell.

The distortion caused when we replace a color c by its quantized value $q(c)$ is called *quantization error*. This error is denoted by $E(c, q(c))$. We can write

$$c = q(c) + E(c, q(c)). \tag{2}$$

A simple method to compute E consists in taking some metric d of the euclidean space $\mathbb{R}^n$, and define $E(c, q(c)) = d(c, q(c))$. Nevertheless, the reader should be aware that perceptual issues are involved in the measure of the quantization distortion. The metric d, in general, does not take that into account. In this paper we will use the *quadratic metric* of the euclidean space: $d(c_i, c_j) = ||c_i - c_j||^2$. This is a good choice from a computational point of view.

Once we know the quantization map q for the image gamut, the quantization of an image is simple: for each pixel color c in the image, we must identify the quantization cell containing c and replace c by the cell's quantization level $q(c)$. Thus, a measure of the quantization error must take into account not only the quantization error of each color, but also the frequency of the occurrence of the color on the image: More frequent colors should produce larger quantization distortions.

If we have N quantization cells $C_1, \ldots, C_N$, a natural measure for the quantization error of an image i is given by

$$E(i) = \sum_{j=1}^{N} \sum_{c \in C_j} F(c)\, E(c, c_j), \tag{3}$$

where $F(c)$ is the frequency of the color c on the image, and c_j is the quantization level of the cell C_j. The frequency $F(c)$ can be easily computed from the *image histogram*, which associates to each color c its frequency of occurrence in the image.

It is important to remark that the computation of the quantization cells and the associated quantization levels are interdependent:

- From the quantization levels, we can compute the quantizations cells by mapping each color to the quantization level nearest to it,

  $$q(c) = c'_i \iff d(c, c'_i) \leq d(c, c'_j), \tag{4}$$

  for all $1 \leq j \leq N$ with $j \neq i$.

- From the quantization cells C_i, $i = 1, \ldots, N$, we compute the quantization level c'_i of each cell C_i in such a way to minimize the quantization error (3) in C_i.

Therefore quantization algorithms should be able to compute either the quantization levels or the quantization cells.

3 Optimal Quantization and Clustering

From (3) we see that the quantization problem is naturally posed as an optimization problem in cluster analysis: The optimal solution should minimize the quantization error $E(i)$ over all possible N-element partitions of the color space.

The rationale behind clustering is to find subsets of a data set that have similar properties. In the case of color image quantization, this framework is used to identify N clusters of colors from the original image that are similar according to the quantization measure. These clusters constitute the quantization cells. For each cluster we compute the quantization level c'_k. Because colors in each cluster are "similar", they can be replaced by their associated quantization level introducing minimal distortion.

The direct solution to optimal clustering implies in a search through the space of all possible cluster configurations to find the one that gives the global minimum. The combinatoric complexity makes this kind of solution intractable. For this reason, clustering methods rely on heuristics in order to find a solution. Non-hierarchical methods start with an initial guess of the cluster configuration that is improved using a relaxation procedure. Hierarchical methods, produce a sequence of cluster configurations generated by means of splitting or merging operations. (See [2].)

In this work we will describe an image quantization algorithm that uses a pairwise clustering optimization technique to obtain quasi-optimal quantization levels for the image gamut.

3.1 Quantization Level of a Color Cluster

We now consider the problem of computing the optimal quantization level associated with some color cluster from an image gamut. The result is stated in the Theorem below:

Theorem 1 *Let $K = \{c_1, c_2, \ldots, c_M\}$ be a cluster of M colors from some color set $C \subset \mathbb{R}^n$ of an image gamut. The optimal quantization level for the cluster K is*

$$c = \frac{1}{\sum_i F_i} \sum_j F_j c_j, \tag{5}$$

where $F_i = F(c_i)$ is the frequency of the color. Moreover, the global quantization error in the cluster is given by

$$E(K) = \frac{1}{(\sum_k F_k)^2} \sum_{j=1}^{M} || \sum_{i=1}^{M} F_i (c_i - c_j) ||^2. \tag{6}$$

The first part of the theorem says that the optimal quantization level of the cluster is the centroid of the cluster.

Proof: Take $E(c, c_j) = ||c - c_j||^2$ in (3), and apply it to the cluster K, to obtain

$$E(c) = \sum_{j=1}^{M} F(c)\, ||c - c_j||^2. \tag{7}$$

This is the quantization error associated to the optimal quantization level c of the cluster. The optimal quantization level c is obtained from the minimum of the function E.

The gradient of E is given by

$$\text{grad}(E) = \sum_{j=1}^{M} 2F(c)(c - c_j). \tag{8}$$

From (8) we obtain the critical point

$$c = \frac{1}{\sum_i F_i} \sum_j F_j c_j. \tag{9}$$

Since the function E is convex, c is indeed a minimal point.

Substituting the minimal point c from (9) into equation (7), we obtain the quantization error in equation (6). This concludes the proof of the Theorem 1.

The following Corollary follows as a particular case of the Theorem.

Corollary 1 *If we have a two color cluster $K = \{c_i, c_j\}$, the optimal quantization level, using the quadratic metric of the euclidean space, is*

$$c = \frac{F_i}{F_i + F_j} c_i + \frac{F_j}{F_i + F_j} c_j, \tag{10}$$

and the associated quantization error is given by

$$E(c_i c_j) = \frac{F_i F_j^2 + F_j F_i^2}{(F_i + F_j)^2}\, ||c_i - c_j||^2 \tag{11}$$

A geometric interpretation of the Corollary provides a good insight into the result. From equation (7) the quantization error is given by the second degree polynomial function

$$E_q(c) = F_i(c - c_i)^2 + F_j(c - c_j)^2.$$

The graph of this function is the arc of parabola shown in Figure 2 (sum of the two dashed arcs of parabola).

Note that the quantization level c is given by the unique minimum point of the parabola. When $F_i = F_j$ que quantization level is the midpoint of the segment $\overline{c_i c_j}$.

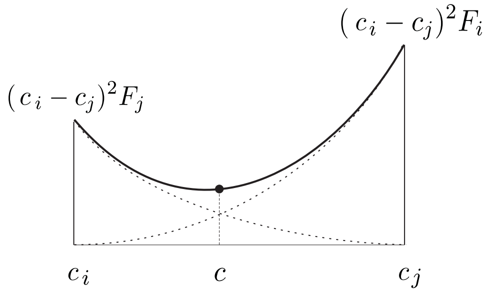

Figure 2: Graph of the quantization error.

4 Pairwise Clustering Quantization

The results from Corollary 1 will now be used to formulate a new color image quantization procedure. The method consists of a relaxation process that computes a sequence of quantization levels by performing locally optimal pairwise cluster merging.

The input to quantization is the finite set C of M colors from the image gamut, $C = \{c_1, c_2, \ldots, c_M\}$. Each color c_i has a frequency $F_i = F(c_i)$. To each color c_i we associate an accumulated quantization error $E(c_i)$, which is initially set to 0.

The image quantization method is composed of the following steps:

1. Compute the image histogram.

2. Using equation (11), compute the quantization error $E(c_i c_j)$ of every two color cluster $\{c_i, c_j\}$ from the color input color set.

3. Choose the two color cluster $K_0 = \{c_i, c_j\}$ from the input color set that minimizes the quantization error $E(c_i c_j)$ computed in the previous step.

4. Using equation (10). Compute the quantization level c of the cluster $K_0 = \{c_i, c_j\}$ chosen in the step 2.

5. Substitute the cluster $K_0 = \{c_i, c_j\}$ by its quantization level c_k. This results in a quantized color set C' with $M - 1$ colors. The frequency $F(c_k)$ of the color c_k is given by the sum of the frequencies of c_i and c_j: $F(c_k) = F_i + F_j$. The accumulated quantization error $E(c_k)$ of the color c_k is given by

$$E(c_k) = E(c_i c_j) + E(c_i) + E(c_j). \tag{12}$$

6. Compute the quantization error for all two cluster colors $\{c_k, c_l\}$, from the quantized color set C'.

7. Use the quantized color set C' as input to step 3 of the algorithm. Repeat steps 3 to 7 until the desired number of quantization levels is obtained.

After the quantization levels are generated, the image quantization is computed using equation (4): For each color c of the image gamut, we quantize it to the closest quantization level using the quadratic metric of the color space.

4.1 Other Strategies to Quantization

We should remark that the quantization strategy described in the previous section consists in optimal collapsing of a two color cluster that constitutes a one-dimensional simplex (c_i, c_j). The optimality of the collapsing procedure is guaranteed by Corollary 1

Different flavors of this quantization method, using the same strategy, consists in collapsing color clusters constituted by simplices of higher dimension (triangles or tetrahedrals). A triangulation of the color space must be performed in order to obtain the clusters to be collapsed.

Certainly, the results will improve if we use higher dimensional simplices. This is because they induce a better spatial correlation in the computation of the quantization error. Unfortunately, the computational complexity increases with simplex dimension, and clever heuristics must be found to make this viable.

5 The Quantization Algorithm

In this section we will describe the implementation of the pairwise color quantization algorithm.

5.1 Data Structures

The algorithm uses two main data structures indexed by color number. The structure Real E[][] for the joint error matrix $E_{i,j}$, and the array c[]:

```
struct c[] {
    Color val;
    Real freq;
    Real err;
}
```

where val is the color value, freq is the color frequency on the image, and err is the accumulated quantization error of color c.

The joint quantization error (11) associated the the two color cluster $\{c_i, c_j\}$, is stored in the matrix entry $E_{i,j}$. From (11) E is a symmetric matrix. Since we will not perform matrix operations with E, we need to store only the elements $E_{i,j}$ for $i < j$, which constitutes a lower triangular matrix.

5.2 Pseudo-Code and Operations

The codification of the algorithm is quite simple:

```
Image Quantize (image, m_levels)
{
    $c$ = compute_histogram(image)
    $E$ = compute_error_matrix($c$)

    while (number_of_colors_in($E$) $>$ m_levels) {
            ($c_i$,$c_j$) = select_color_pair($E$)
            $c_k$ = merge_color_pair($c_i$, $c_j$)
            replace_colors($c_i$, $c_j$) by $c_k$
    }
    apply_quantization(image, c)
}
```

The function compute_histogram computes the frequency of each color in the input image, and selects the set of colors with frequency freq > 0. In order to reduce the computation involved, the image is uniformly quantized to 15 bits before computing the histogram.

The function compute_error_matrix computes $E_{i,j}$ using equation (11).

The function select_color_pair simply selects the pair of colors (c_i, c_j) with minimum joint quantization error $E_{i,j}$ in the matrix E.

The function merge_color_pair computes a new quantized color c_k to replace the cluster $\{c_i, c_j\}$, using equation (10).

The function replace_colors substitutes the two color cluster $\{c_i, c_j\}$ by the quantized color c_k. This is done in three stages: First $F_k = F_i + F_j$ and $E_k = E_{i,j} + E_i + E_j$ are computed; Second, the entries for colors c_i and c_j are deleted from the array c and from the matrix E; Third, a new entry for color c_k is added to both c and E.

Function replace_colors is central to the algorithm, and its operation can be stated in terms of simple operations with matrix E: Deleting a color c_j is equivalent to remove the jth row and column from E. If the matrix E has order n (n colors are being processed), adding a new color c_k, is equivalent to append the $(n+1)-$th row at the end of the matrix E. This row stores the joint errors of pairs of colors formed by the new color c_k with all other colors c_l, $l = 1 \ldots n$.

6 Results

To demonstrate the results of our color image quantization algorithm we have selected two 24-bit RGB images: one, “Fish”, is a computer-generated scene; and the other,“Parrots”, is a photograph of a natural scene. These images are shown in Figure 3.

To be able to visualize the color clusters generated by our method, we have selected a detail of “Parrots” with the Blue component set to zero (see Figure 4). In that way the color space of the image is restricted to the Red–Green plane. The set of image colors are shown in Figure 4, where the colored dots have an area proportional to the frequency of their associated color.

The image in 4 is quantized to 16 colors using our method and the color clusters produced are shown in Figure 5. Note how the color space partition is well adapted to the set of image colors.

6.1 Visual Comparison

Figures 6 to 9 present color quantizations of the original images. The results generated by our method are compared with those from the median-cut algorithm [1], which was chosen due to its widespread use in imaging applications.

The images have a spatial resolution of 300x200 pixels and will be quantized, respectively, to 256 and 16 colors using both the pairwise clustering algorithm (PCA), and the median–cut algorithm (MCA). The results of PCA are shown on the left side, and the results of MCA on the right side. Below each quantized image we show its normalized quantization error.

Figure 6 shows a 256-color quantization of the ”Fish”. One of the most import characteristic of the PCA is the preservation of color visual information. Note that in the fish’s fin there are some blue points that are destroyed by the MCA, but this information was preserved by our algorithm (PCA).

The ”Fish” is quantized to 16 colors in Figure 7. Looking at both images we observe that the PCA has chosen colors that are closer to the original colors. Note that our algorithm (PCA) spreads the quantization error evenly, while the MCA leads to areas with large errors, for example the highlight areas.

In Figure 8, images of the ”Parrots” quantized to 256 colors by PCA and MCA look quite similar. A more extreme quantization to 16 colors is shown in Figure 9. In this last case, MCA produced colors that are not present in the original image, specially in the background where there are red areas that do not exist in the original image.

It’s important to emphasize that the PCA spreads the quantization error throughout the entire image while the MCA tends to concentrate its error in some image regions.

6.2 Numerical Comparison

Now we compare PCA with other popular quantizations algorithms: uniform quantization, populosity algorithm [1], median-cut [1], local K-means [3], variance-based [4] and octree [5].

For a numeric comparison we employ the quadratic metric. The results of the tests are in Table 1, which gives the total quantization error divided by the number of pixels in the image (i.e. the RMS error) for all algorithms tested. Note that our algorithm is consistently superior to all others.

Algorithm	Fish		Parrots	
No. of colors	16	256	16	256
Uniform	90.35	27.61	105.56	29.56
Popularity	34.10	6.56	54.26	9.20
Median cut	17.67	6.42	22.23	7.27
Local K-means	18.02	7.50	25.67	9.43
Variance-based	16.50	6.85	20.18	7.16
Octree	18.22	5.52	21.11	6.66
Pairwise cluster	15.16	5.54	18.46	6.24

Table 1: Comparisons of quantization methods.

7 Conclusion

We have proposed a new color image quantization method based on iterative clustering techniques. The algorithm is simple to implement and gives very good results.

The pairwise clustering algorithm has computational complexity proportional to the square of the number of input colors. As a future work we are making efforts to improve the efficiency of the algorithm using a cluster split–merge strategy.

Acknowledgements

This research has been developed in the laboratory of the VISGRAF project at IMPA. This project is sponsored by research grants from CNPq, FAPERJ, FINEP, and IBM Brasil.

The voronoi diagram in Figure 5 was created using a program by Steve Fortune [6]. We thank Luiz Henrique Figueiredo for helping to generate this figure.

References

[1] P. Heckbert. Color image quantization for frame buffer display. *Computer Graphics*, 16(3):297–307, July 1982.

[2] H. Spath. *Cluster Analysis Algorithms*. Halsted Press, 1980.

[3] O. Verevka and J. Buchanan. Local k-means algorithm for color image quantization. (unpublished).

[4] P. Prusinkewicz S. J. Wan and S. K. M. Wong. Variance-based color image quantization for frame buffer display. *Color Research and Aplication*, 15:52–58, February 1990.

[5] M. Gervautz and W. Purgathofer. A simple method for color quantization: octree quantization. In *Proceedings of CGI 88*, pages 219–231, 1988.

[6] Steve J. Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2:153–174, 1987.

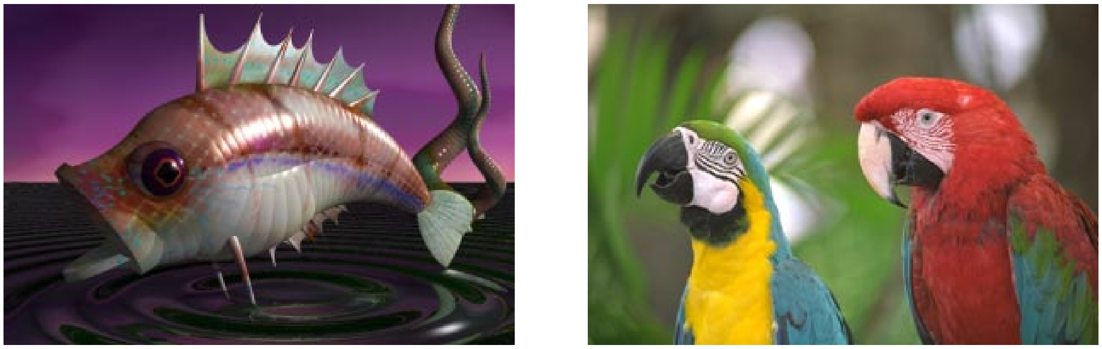

Figure 3: Original RGB images: Fish (left) and Parrots (right)

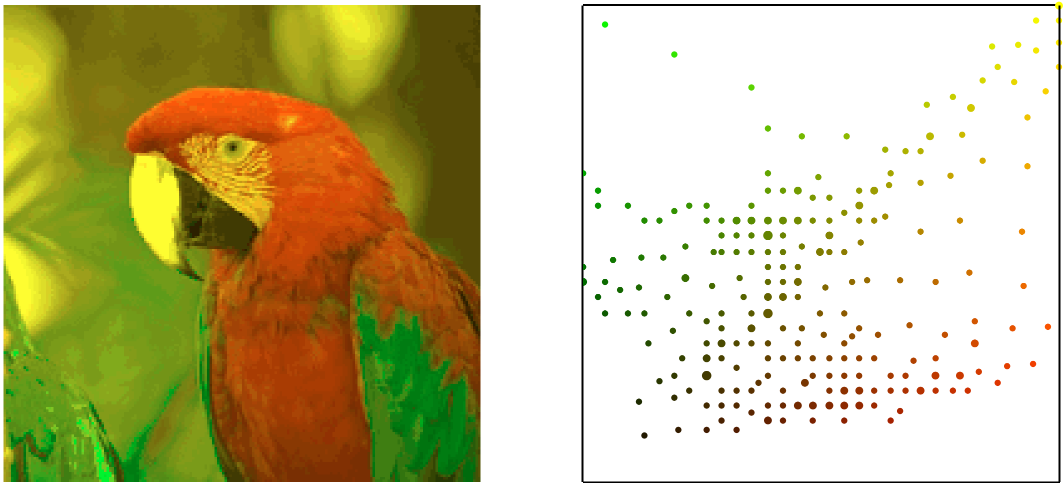

Figure 4: Detail of Parrots projected to Red-Green color plane (left). Image Gamut (right). Area of dots is proportional to color frequency

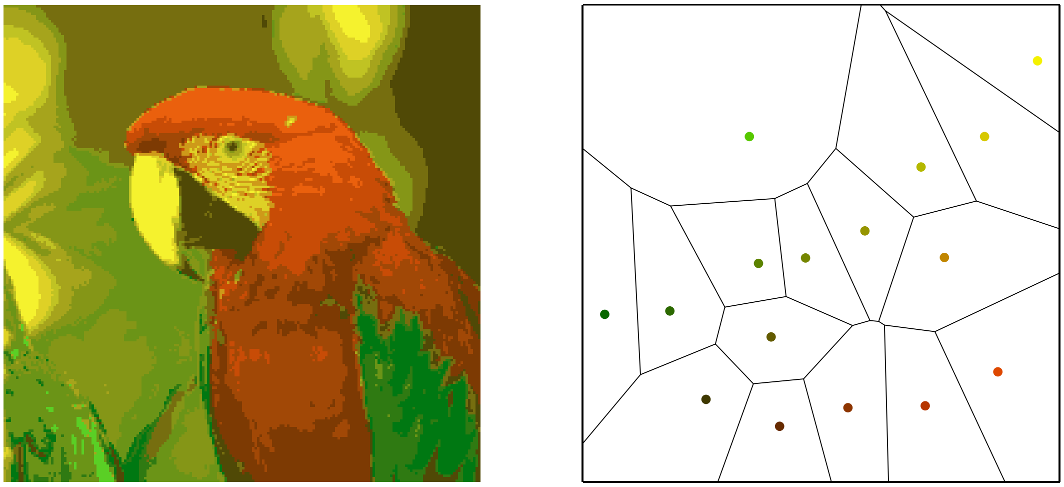

Figure 5: Red-Green image quantized to 16 colors (left). Color clusters and quantization levels (right)

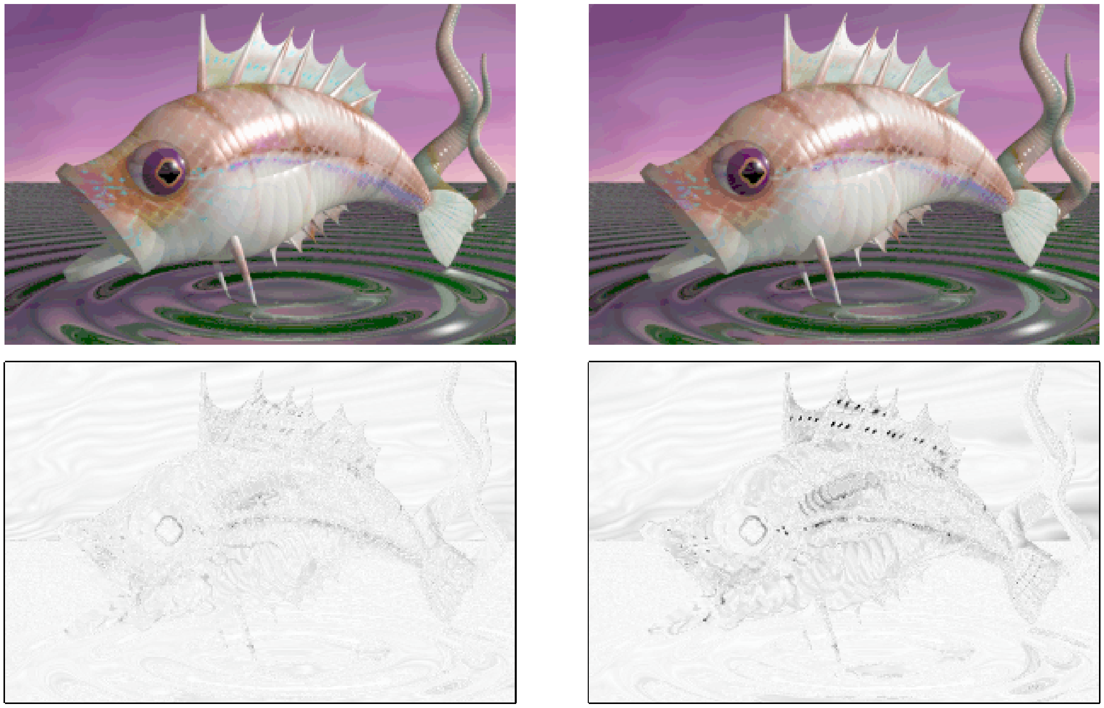

Figure 6: Fish quantized to 256 colors. Pairwise Clustering (left). Median Cut (right).

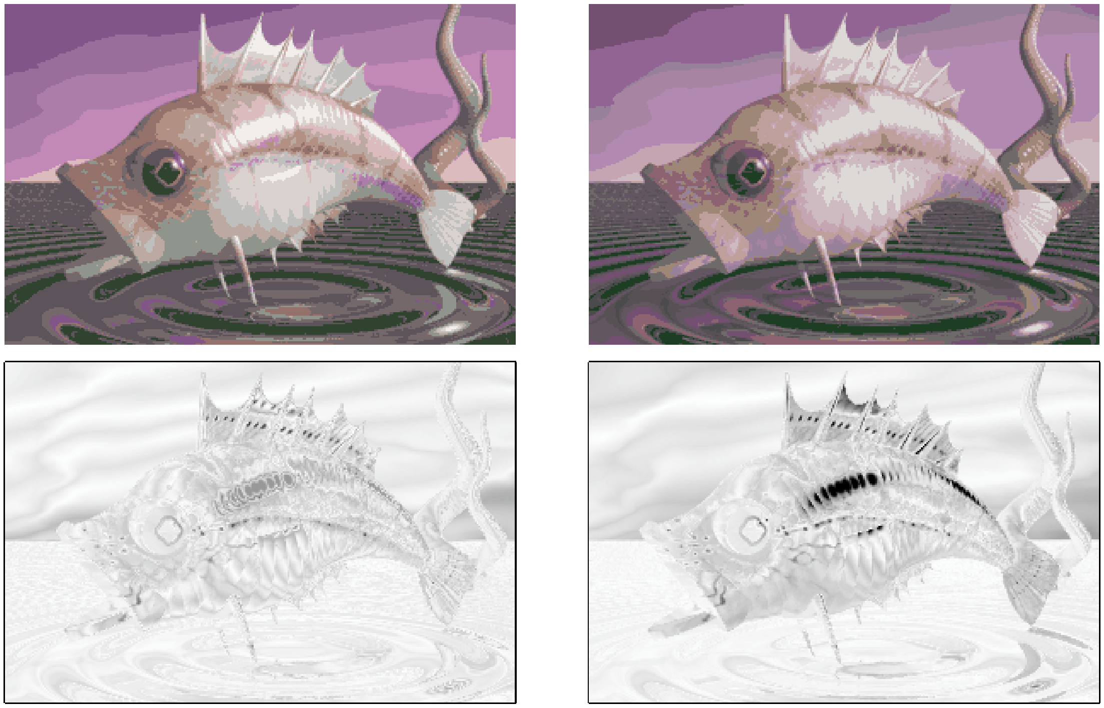

Figure 7: Fish quantized to 16 colors. Pairwise Clustering (left). Median Cut (right).

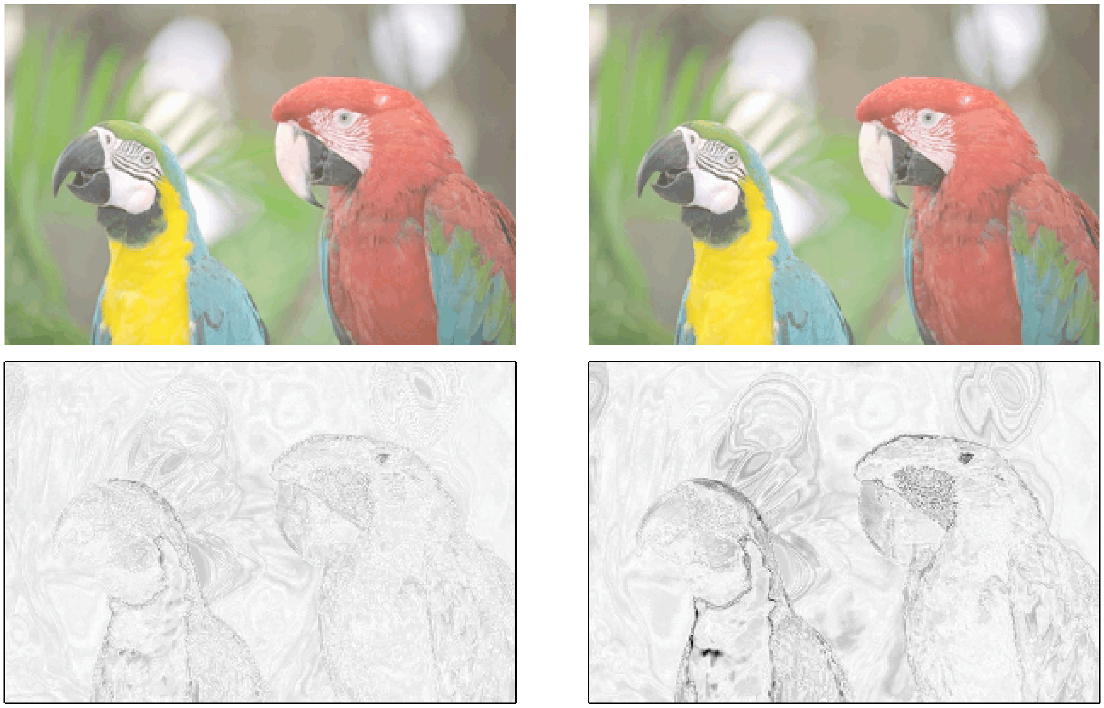

Figure 8: Parrots quantized to 256 colors. Pairwise Clustering (left). Median Cut (right).

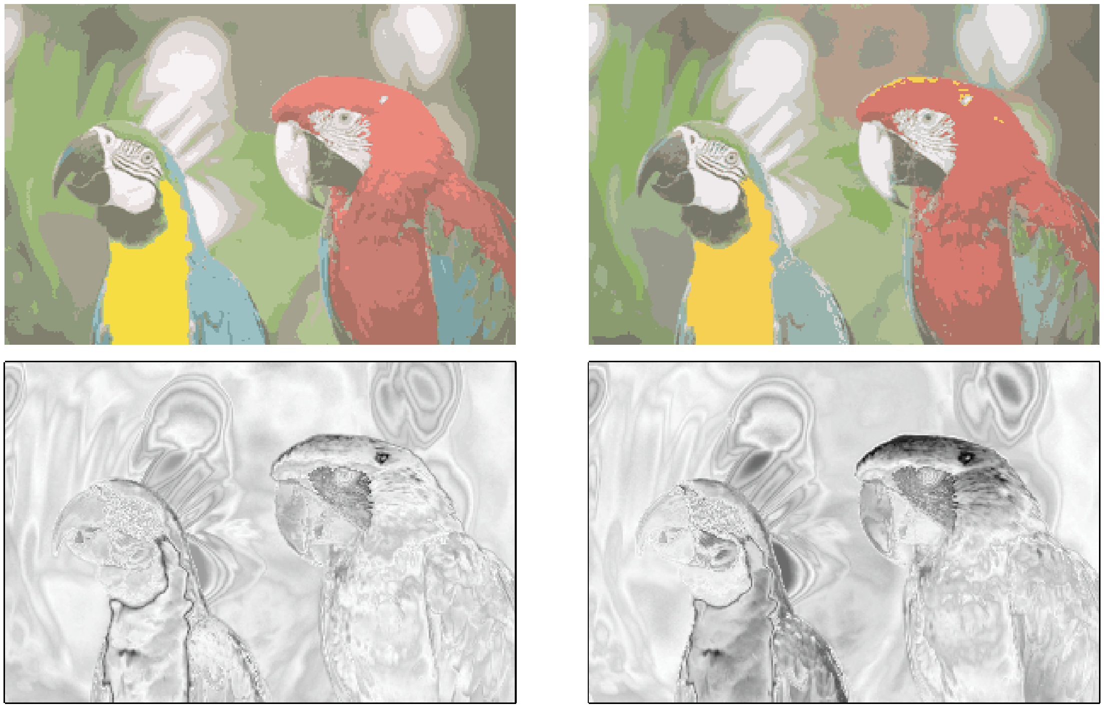

Figure 9: Parrots quantized to 16 colors. Pairwise Clustering (left). Median Cut (right).