COLOR QUANTIZATION USING C-MEANS CLUSTERING ALGORITHMS

M. Emre Celebi 1, Quan Wen 2, and Juan Chen 2

1 Department of Computer Science, Louisiana State University, Shreveport, USA
2 School of Computer Science & Eng., Univ. of Elec. Sci. & Tech. of China, Chengdu, P.R. China

ABSTRACT
Color quantization is an important operation with many applications in graphics and image processing. Most quantization methods are essentially based on data clustering algorithms. Recent studies have demonstrated the effectiveness of hard c-means (k-means) clustering algorithm in this domain. Other studies reported similar findings pertaining to the fuzzy c-means algorithm. Interestingly, none of these studies directly compared the two types of c-means algorithms. In this study, we implement fast and exact variants of the hard and fuzzy c-means algorithms with several initialization schemes and then compare the resulting quantizers on a diverse set of images. The results demonstrate that fuzzy c-means is significantly slower than hard c-means, and that with respect to output quality the former algorithm is neither objectively nor subjectively superior to the latter.

1. INTRODUCTION

True-color images typically contain thousands of colors, which makes their display, storage, transmission, and processing problematic. For this reason, color quantization (reduction) is commonly used as a preprocessing step for various graphics and image processing tasks. In the past, color quantization was a necessity due to the limitations of the display hardware, which could not handle over 16 million possible colors in 24-bit images. Although 24-bit display hardware has become more common, color quantization still maintains its practical value [1]. Modern applications of color quantization in graphics and image processing include: (i) compression, (ii) segmentation, (iii) text localization/detection, (iv) color-texture analysis, (v) watermarking, (vi) non-photorealistic rendering, and (vii) content-based retrieval.

The process of color quantization is mainly comprised of two phases: palette design (the selection of a small set of colors that represents the original image colors) and pixel mapping (the assignment of each input pixel to one of the palette colors). The primary objective is to reduce the number of unique colors, \( N' \), in an image to \( C, C \ll N' \), with minimal distortion. In most applications, 24-bit pixels in the original image are reduced to 8 bits or fewer. Since natural images often contain a large number of colors, faithful representation of these images with a limited size palette is a difficult problem.

Color quantization methods can be broadly classified into two categories [1]: image-independent methods that determine a universal (fixed) palette without regard to any specific image, and image-dependent methods that determine a custom (adaptive) palette based on the color distribution of the images. Despite being very fast, image-independent methods usually give poor results since they do not take into account the image contents. Therefore, most of the studies in the literature consider only image-dependent methods, which strive to achieve a better balance between computational efficiency and visual quality of the quantization output.

Numerous image-dependent color quantization methods have been developed in the past three decades. These can be categorized into two families: preclustering methods and postclustering methods [1]. Preclustering methods are mostly based on the statistical analysis of the color distribution of the images. Divisive preclustering methods start with a single cluster that contains all \( N \) image pixels. This initial cluster is recursively subdivided until \( C \) clusters are obtained. Well-known divisive methods include median-cut [2], octree [3], variance-based method [4], and greedy orthogonal bipartitioning method [5]. On the other hand, agglomerative preclustering methods [6, 7] start with \( N \) singleton clusters each of which contains one image pixel. These clusters are repeatedly merged until \( C \) clusters remain. In contrast to preclustering methods that compute the palette only once, postclustering methods first determine an initial palette and then improve it iteratively. Essentially, any data clustering method can be used for this purpose. Since these methods involve iterative or stochastic optimization, they can obtain higher quality results when compared to preclustering methods at the expense of increased computational time. Clustering algorithms adapted to color quantization include hard c-means [8, 9], competitive learning [10, 11, 12], fuzzy c-means [13, 14, 15, 16], and self-organizing maps [17, 18].

In this paper, we compare the performance of hard and fuzzy c-means algorithms within the context of color quantization. The rest of the paper is organized as follows. Section 2 reviews the notions of hard and fuzzy partitions and gives
an overview of the HCM and FCM algorithms. Section 3 describes the experimental setup and compares the HCM and FCM variants on a set of test images. Finally, Section 4 gives the conclusions.

2. COLOR QUANTIZATION USING C-MEANS CLUSTERING ALGORITHMS

2.1. Hard vs. Fuzzy Partitions

Given a data set \( X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^D \), a real matrix \( U = [u_{ik}]_{C \times N} \) represents a hard \( C \)-partition of \( X \) if and only if its elements satisfy three conditions [19]:

\[
\begin{align*}
  u_{ik} &\in \{0, 1\} \quad 1 \leq i \leq C, \quad 1 \leq k \leq N \\
  \sum_{i=1}^{C} u_{ik} & = 1 \quad 1 \leq k \leq N \\
  0 &< \sum_{k=1}^{N} u_{ik} < N \quad 1 \leq i \leq C 
\end{align*}
\]  

(1)

For obvious reasons, \( U \) is often called a partition or membership matrix. The concept of hard \( C \)-partition can be generalized by relaxing the first condition in Eq. (1) as \( u_{ik} \in [0, 1] \) in which case the partition matrix \( U \) is said to represent a fuzzy \( C \)-partition of \( X \).

2.2. Hard C-Means Clustering Algorithm

The HCM algorithm is inarguably one of the most widely used methods for data clustering. HCM attempts to generate optimal hard \( C \)-partitions of \( X \) by minimizing the following objective functional:

\[
J(U, V) = \sum_{k=1}^{N} \sum_{i=1}^{C} u_{ik} (d_{ik})^2 
\]  

(2)

where \( U \) is a hard partition matrix as defined in §2.1, \( V = \{v_1, v_2, \ldots, v_C\} \in \mathbb{R}^D \) is a set of \( C \) cluster representatives (centers), e.g. \( v_i \) is the center of hard cluster \( U_i \), \( \forall i \), and \( d_{ik} \) denotes the Euclidean (\( L_2 \)) distance between input vector \( x_k \) and cluster representative \( v_i \), i.e. \( d_{ik} = \|x_k - v_i\|_2 \).

This problem is known to be NP-hard [20], but a heuristic method developed by Lloyd [21] offers a simple solution. Lloyd’s algorithm starts with \( C \) arbitrary centers, typically chosen uniformly at random from the data points. Each point is then assigned to the nearest center, and each center is recalculated as the mean of all points assigned to it. These two steps are repeated until a predefined termination criterion is met.

From a clustering perspective HCM has the following advantages: (1) It is conceptually simple, versatile, and easy to implement, (2) It has a time complexity of \( \mathcal{O}(NC^2) \) per iteration, and (3) It is guaranteed to terminate. Due to its gradient descent nature, HCM often converges to a local minimum of its objective functional and its output is highly sensitive to the selection of the initial cluster centers. From a color quantization perspective, HCM has two additional drawbacks. First, despite its linear time complexity, the iterative nature of the algorithm renders the palette generation phase computationally expensive. Second, the pixel mapping phase is inefficient, since for each input pixel a full search of the palette is required to determine the nearest color.

We have recently proposed a fast and exact HCM variant called Weighted Sort-Means (WSM) that utilizes data reduction and accelerated nearest neighbor search [9]. When initialized with a suitable preclustering method, WSM has been shown to outperform a large number of classic and state-of-the-art quantization methods. In the experiments, WSM was used in place of HCM. For uniformity, in the remainder of this paper, we will refer to WSM as HCM since both algorithms give numerically identical results.

2.3. Fuzzy C-Means Clustering Algorithm

Fuzzy c-means (FCM) [19] is a generalization of HCM in which points can belong to more than one cluster. FCM attempts to generate optimal fuzzy \( C \)-partitions of \( X \) by minimizing the following objective functional:

\[
J_m(U, V) = \sum_{k=1}^{N} \sum_{i=1}^{C} u_{ik}^m (d_{ik})^2 
\]  

(3)

where the parameter \( 1 \leq m < \infty \) controls the degree of membership sharing between fuzzy clusters in \( X \).

As in the case of HCM, FCM is based on an alternating minimization procedure. At each iteration, the fuzzy \( C \)-partition matrix \( U \) is updated by

\[
u_{ik} = \left[ \frac{\sum_{j=1}^{C} (d_{jk})^{2/(m-1)}}{\sum_{j=1}^{C} (d_{jk})} \right]^{1/m}.
\]  

(4)

which is followed by the update of the prototype matrix \( V \) by

\[
v_i = \left( \sum_{k=1}^{N} (u_{ik})^m x_k \right) / \left( \sum_{k=1}^{N} (u_{ik})^m \right).
\]  

(5)

In color quantization applications, in order to map each input color to the nearest (most similar) palette color, the membership values should be defuzzified upon convergence.

A naïve implementation of the FCM algorithm has a complexity of \( \mathcal{O}(NC^2) \) per iteration, which is quadratic in the number of clusters. In the experiments, a linear complexity formulation, i.e. \( \mathcal{O}(NC) \), described in [22] was used. In order to take advantage of the peculiarities of color image data (presence of duplicate samples, limited range, and sparsity), the same data reduction strategy used in WSM was incorporated into FCM.
3. EXPERIMENTAL RESULTS AND DISCUSSION

Four publicly available, true-color images from the Kodak Lossless True Color Image Suite [23] were used in the experiments. The effectiveness of a quantization method was quantified by the commonly used Mean Squared Error (MSE) measure [1]. Computational efficiency was measured by CPU time in milliseconds (gcc v4.4.3, Intel Xeon E5520 2.26GHz). The time figures were averaged over 20 runs.

The following well-known preclustering methods were used in the experiments: median-cut (MC), octree (OCT), variance-based method (WAN), and greedy orthogonal bipartitioning method (WU). Four variants of HCM/FCM, each one initialized with a different preclustering method were tested.

Table 1 compares the effectiveness of the HCM and FCM variants on the test images. For a given number of colors $C$ ($C \in \{32, 64, 128, 256\}$), preclustering method $P$ ($P \in \{MC, OCT, WAN, WU\}$), and image $I$, the column labeled as ‘Init’ contains the MSE between $I$ and $\hat{I}$ (the image obtained by reducing $I$ to $C$ colors using $P$), whereas the one labeled as ‘HCM’ (‘FCM’) contains the MSE obtained by the HCM (FCM) algorithm when it is initialized by $P$. Following [14], $m = 1.25$ was used in FCM. Similarly, Table 2 compares the efficiency of the HCM and FCM variants. The following observations are in order:

- Both HCM and FCM reduce the quantization distortion regardless of the initialization method used. However, the percentage of MSE reduction is more significant for some initialization methods than others.
- HCM is more effective than FCM.
- HCM is significantly faster than FCM (see the column labeled as ‘F/H’ in Table 2). This is because HCM uses hard memberships, which makes possible various computational optimizations that do not affect accuracy of the algorithm.

Figure 1 shows sample quantization results for the Motocross image. Since WU is the most effective initialization method, only the outputs of HCM/FCM variants that use WU are shown. It can be seen that WU is unable to represent the color distribution of certain regions of the image (fenders of the leftmost and rightmost dirt bikes, helmet of the driver of the leftmost dirt bike, grass, etc.) In contrast, HCM and FCM perform significantly better in allocating representative colors to these regions.

4. CONCLUSIONS

In this paper, hard and fuzzy c-means clustering algorithms were compared within the context of color quantization. Fast and exact variants of both algorithms with several initialization schemes were compared on a diverse set of publicly available test images. The results indicate that fuzzy c-means does not seem to offer any advantage over hard c-means. Furthermore, due to the intensive membership calculations involved, fuzzy c-means is significantly slower than hard c-means, which makes it unsuitable for time-critical applications.
## Table 2. CPU time comparison of the quantization methods

<table>
<thead>
<tr>
<th>C</th>
<th>HCM</th>
<th>PCM</th>
<th>FH</th>
<th>HCM</th>
<th>PCM</th>
<th>FH</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>MC</td>
<td>88</td>
<td>2664</td>
<td>36</td>
<td>84</td>
<td>1179</td>
</tr>
<tr>
<td></td>
<td>OCT</td>
<td>80</td>
<td>1883</td>
<td>24</td>
<td>110</td>
<td>4139</td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>45</td>
<td>3406</td>
<td>77</td>
<td>60</td>
<td>4261</td>
</tr>
<tr>
<td></td>
<td>WU</td>
<td>50</td>
<td>1976</td>
<td>40</td>
<td>60</td>
<td>4547</td>
</tr>
<tr>
<td>64</td>
<td>MC</td>
<td>39</td>
<td>10036</td>
<td>180</td>
<td>101</td>
<td>29081</td>
</tr>
<tr>
<td></td>
<td>OCT</td>
<td>97</td>
<td>5045</td>
<td>52</td>
<td>130</td>
<td>10154</td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>62</td>
<td>9350</td>
<td>151</td>
<td>94</td>
<td>4261</td>
</tr>
<tr>
<td></td>
<td>WU</td>
<td>54</td>
<td>4228</td>
<td>78</td>
<td>71</td>
<td>6361</td>
</tr>
<tr>
<td>128</td>
<td>MC</td>
<td>108</td>
<td>20269</td>
<td>189</td>
<td>156</td>
<td>49930</td>
</tr>
<tr>
<td></td>
<td>OCT</td>
<td>141</td>
<td>12700</td>
<td>90</td>
<td>180</td>
<td>22410</td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>89</td>
<td>22871</td>
<td>258</td>
<td>125</td>
<td>17472</td>
</tr>
<tr>
<td></td>
<td>WU</td>
<td>76</td>
<td>19058</td>
<td>168</td>
<td>113</td>
<td>15604</td>
</tr>
<tr>
<td>256</td>
<td>MC</td>
<td>124</td>
<td>35372</td>
<td>285</td>
<td>120</td>
<td>47186</td>
</tr>
<tr>
<td></td>
<td>OCT</td>
<td>120</td>
<td>9787</td>
<td>82</td>
<td>130</td>
<td>14311</td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>86</td>
<td>10875</td>
<td>127</td>
<td>103</td>
<td>19432</td>
</tr>
<tr>
<td></td>
<td>WU</td>
<td>84</td>
<td>9145</td>
<td>110</td>
<td>95</td>
<td>11016</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>HCM</th>
<th>PCM</th>
<th>FH</th>
<th>HCM</th>
<th>PCM</th>
<th>FH</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>MC</td>
<td>56</td>
<td>5591</td>
<td>100</td>
<td>55</td>
<td>6067</td>
</tr>
<tr>
<td></td>
<td>OCT</td>
<td>81</td>
<td>2618</td>
<td>33</td>
<td>82</td>
<td>1992</td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>42</td>
<td>2240</td>
<td>53</td>
<td>45</td>
<td>1934</td>
</tr>
<tr>
<td></td>
<td>WU</td>
<td>42</td>
<td>2111</td>
<td>51</td>
<td>41</td>
<td>1927</td>
</tr>
<tr>
<td>64</td>
<td>MC</td>
<td>62</td>
<td>10508</td>
<td>171</td>
<td>77</td>
<td>14165</td>
</tr>
<tr>
<td></td>
<td>OCT</td>
<td>99</td>
<td>9091</td>
<td>92</td>
<td>100</td>
<td>6431</td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>58</td>
<td>5413</td>
<td>94</td>
<td>59</td>
<td>6540</td>
</tr>
<tr>
<td></td>
<td>WU</td>
<td>53</td>
<td>3887</td>
<td>73</td>
<td>62</td>
<td>5745</td>
</tr>
<tr>
<td>128</td>
<td>MC</td>
<td>124</td>
<td>33792</td>
<td>285</td>
<td>120</td>
<td>47186</td>
</tr>
<tr>
<td></td>
<td>OCT</td>
<td>120</td>
<td>9787</td>
<td>82</td>
<td>130</td>
<td>14311</td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>86</td>
<td>10875</td>
<td>127</td>
<td>103</td>
<td>19432</td>
</tr>
<tr>
<td></td>
<td>WU</td>
<td>84</td>
<td>9145</td>
<td>110</td>
<td>95</td>
<td>11016</td>
</tr>
<tr>
<td>256</td>
<td>MC</td>
<td>368</td>
<td>63209</td>
<td>172</td>
<td>403</td>
<td>84079</td>
</tr>
<tr>
<td></td>
<td>OCT</td>
<td>291</td>
<td>30560</td>
<td>105</td>
<td>279</td>
<td>31042</td>
</tr>
<tr>
<td></td>
<td>WAN</td>
<td>223</td>
<td>28113</td>
<td>126</td>
<td>238</td>
<td>33780</td>
</tr>
<tr>
<td></td>
<td>WU</td>
<td>226</td>
<td>19480</td>
<td>86</td>
<td>216</td>
<td>27107</td>
</tr>
</tbody>
</table>

5. REFERENCES


