Clustered based color reduction
- Improvements and tips -

Miti Ruchanurucks
The University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505 Japan
E-mail: miti@cvl.iis.u-tokyo.ac.jp

Abstract This paper presents simple yet powerful improvements in color reduction field, targeting interactive high-quality applications. Here maximum distance clustering (MDC) is used to initialize K-means clustering, which eliminates the drawback of clustering-based color reduction that tends to ignore colors with a small number of pixels. Maximum distance clustering’s speed problem due to the problem of dimensionality is solved by using a proposed sub-optimal algorithm. Furthermore, it is shown that behavior of MDC and K-means in RGB color space is different from that in CIELAB color space. Our improvements enable simple algorithms like MDC and K-means to outperform many existing color reduction algorithms.

Keyword Image processing, image coding, image color analysis, image generation, clustering method

1. INTRODUCTION

Usually, digital color images consist of up to 16 million different colors in a 24-bit color space. However, in many applications, namely compression, presentation, and transmission, it is preferred to have as small a number of colors as possible. Color reduction is a process that transforms a full color image to an image with a smaller number of colors, by grouping similar colors and replacing them with a representative color.

Presently, several paradigms for color reduction have been proposed. The first scheme processes colors by splitting the color space into smaller regions. Among the methods for accomplishing this are median-cut [1], octree [2], and variance-based algorithm [3]. However, their disadvantage is that the resulting image often contains regions with colors largely different from the original ones, when viewed by human eyes. The second paradigm falls into the domain of clustering. K-means [4] and [5], C-means [6], and fuzzy C-means [7] are among the practical clustering-based color reduction methods. A general drawback of clustering is that it tends to ignore colors with a small number of pixels and joins them to a neighboring larger cluster, which lessens the contrast in the output image. Another more complex approach makes use of a neural network for better classification results, for example, the Kohonen network [8], adaptive color reduction [9], and Neural Gas [10]. Nevertheless, exploitation of the neural net is time consuming and difficult to implement.

The fourth paradigm tries to enhance the quality of the output image by considering spatial characteristics, for example, dithered color quantization [11]. The method can produce good quality images by performing color reduction along with dithering at the same time. However, this consumes more time than is acceptable from an interactive standpoint. The final scheme is region-based color segmentation, such as [12], which automatically finds color segments based on the minimum acceptable size of a region specified by a user or [13], which allows a user to specify not only a color difference threshold but also the spatial radius of a filter and the minimum acceptable size of a region for a given color. However, this approach often cannot reduce the number of colors effectively, and, instead, is applied in the field of region segmentation. Ultimately, although many techniques are available, an acceptable interactive method that can produce a high-quality result is still needed.

This paper proposes improvements in a color reduction scheme based on clustering based approach of [14] which incorporates two clustering methods, maximum distance clustering (MDC) and K-means. It shows that using MDC + an iteration of K-means, in RGB color space, the result is better than using K-means alone. This results in high-quality, well-contrasted output images; and even the reduction in the number of colors is very low. We extends [14] by solving speed problem of MDC using a proposed sub-optimal algorithm. Then it is shown that behavior of clustering schemes in CIELAB color space is different from that in RGB color space. Another objective of this paper is an algorithm that anyone could easily implement, and this is already achieved since our improvement is based on well-known and easy algorithms, MDC and K-means.

For benchmarking, we use Photoshop’s perceptual-based palette generation [15], region-based color segmentation of watershed [12], and clustering-based method of normal K-means. It will be shown that, in RGB color space, although MDC + an iteration K-means performs better than K-means, however, in CIELAB color space, both approaches are comparable if K-means is performed forward and backward on image coordinate.
This key problem of high-quality interactive color reduction is what our method can directly address. In section 2, the incorporation of MDC and K-means is described. Our discovery is shown in both section 2 and section 3. Finally, section 4 contains our conclusions.

2. MDC + 1 K-MEANS

In order to solve the contrast problem of clustering-based color reduction, maximum distance clustering (MDC), considered a weak clustering method, is initially used to specify the highest contrasted colors. The set of maximum contrast colors is then used to initialize K-means clustering. K-means clustering is robust and widely used in various fields of research. The drawback is that it tends to ignore colors with a small number of pixels and joins them to a neighboring larger cluster, which lessens the contrast in the output image. Hence, [14] propose a method that weights the contrasted colors obtained by MDC and the statistically calculated colors obtained by K-means.

To capture the maximum contrast of colors of an image, MDC is applied to the image in the RGB color space. The first iteration starts by sampling a pixel from the input image and then identifying a color with the largest Euclidean distance from the sampled pixel’s color. For all consecutive iteration, the criterion for a new color is (1).

$$d_{\text{max},k+1} = \min_k \max_{i,j} \left\| c_k - p_{i,j} \right\|$$

where \(d_{\text{max},k+1}\) is the maximum distance of cluster \(k+1\), \(k\) is the number of existing cluster, \(i,j\) is the coordinate of a pixel, \(c_k\) is the existing cluster’s colors, and \(p_{i,j}\) represents colors at each coordinate, and is a candidate for a new cluster.

Since the colors would be used to initialize K-means clustering, the algorithm continues until the number of clusters equals the desired number of colors, or there is no other candidate color. In other words, the method requires running \(k\) iterations on an image to find \(k\) initial cluster means.

The derived highest contrasted colors are used as initial means for K-means clustering. Then, each new piece of data is used to compute the new mean of the closest cluster derived from (2).

$$\min_k \left\| c_k - p_{i,j} \right\|$$

In order to prevent K-means from dominating the clustering process, it must not be run until converged. Practically, running K-means for only one iteration gives the best result. Finally, each pixel of an image is rendered based on the closest derived cluster mean.

2.1 Sub-optimal MDC

Since it is preferable that color reduction returns the output image as fast as possible, MDC is the bottleneck of [14] as it consumes a considerable amount of time when the number of colors is high (see table 1). This is due to the fact that searching for one maximum distance cluster requires an iteration search on an image. So we propose a sub-optimal MDC.

The algorithm starts similarly by identifying a color with the largest Euclidean distance from a sampled pixel’s color. The difference is that every new cluster color is derived within the second iteration search by a criterion explained as follows. First, calculate a minimum Euclidean distance of a pixel from the existing cluster(s) using (3).

$$d_{\text{min},i,j} = \min_k \left\| c_k - p_{i,j} \right\|$$

if \(d_{\text{min},i,j} > d_{\text{max},k}\) for any cluster \(k\),
replace the existing cluster with \(p_{i,j}\).
else if \(d_{\text{min},i,j} \neq 0\) & \(d_{\text{min},i,j} \leq d_{\text{max},k}\) for all clusters,
generate a new cluster with value equal to \(p_{i,j}\) if the present number of clusters is still lower than the desired number.

Using this new algorithm, running sub-optimal MDC consumes approximately the same time as running an iteration of K-means. In other words, running sub-optimal MDC + an iteration of K-means consumes time approximately equal to that of two iterations K-means.

2.2 Experimental result

As shown in figure 1, it can be seen that the (sub-optimal) MDC + an iteration of K-means outperforms the region-based algorithm of [12], the commercial perceptual-based color reduction by Photoshop [15], and the normal K-means clustering. Note that we discover that running K-means using the same order of input will result in even lower contrasted image, so K-means, here, is performed forward for one iteration and backward for the next iteration, and vice versa, on image coordinate domain.

Thus, using sub-optimal MDC + 1 K-means is preferable to running K-means alone since the former give better result in shorter time, in RGB color space.
Figure 1 Comparison of color reduction tools when performing color reduction to 16 colors in RGB color space.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time complexity</th>
<th>16 colors</th>
<th>64 colors</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>$O(N \cdot k \cdot i)$</td>
<td>2.33</td>
<td>7.64</td>
</tr>
<tr>
<td>MDC + an iteration of K-means</td>
<td>$O(N \cdot k^2) + O(N \cdot k)$</td>
<td>1.21</td>
<td>15.83</td>
</tr>
<tr>
<td>Sub-optimal MDC + an iteration of K-means</td>
<td>$O(N \cdot k) + O(N \cdot k)$</td>
<td>0.29</td>
<td>1.03</td>
</tr>
</tbody>
</table>

Table 1 Algorithm time comparison. $N$ is the image size (here 640x480), $k$ is the number of colors, and $i$ is the number of iterations until convergence (here 15). Time is in seconds. Average is based on 20 images using a laptop with 1.8 GHz CPU.

### 3. PERFORMANCE IN CIELAB COLOR SPACE

According to [14], using MDC with K-means gives better result than using K-means alone. In this paper, we discover that such a claim is trivial if the clustering is performed on CIELAB color space, instead of other color spaces like, e.g., RGB.

CIELAB is designed to produce a color that is more perceptually linear than other color space, meaning that a change of the same amount in color value should result in a change of about the same visual importance. Nevertheless, it might be assumed that the comparison result in former section should hold even the color space is changed from RGB to CIELAB. Surprisingly, it is not.

This paper shows that, using CIELAB, two powerful yet easy-to-implement color reduction methods can be achieved. The best choice is the sub-optimal MDC + an iteration of K-means. The second one is running K-means for two iterations, forward and backward.

#### 3.1 Experimental result

Comparing figure 1 and 2, it can be seen that if CIELAB is used, both K-means and sub-optimal MDC + an iteration of K-means can generate images that can preserve contrast and objects in scene well.
4. CONCLUSION

Points in such color domain does not provide meaningful measurement.

We also tested another color space, HSL. The results are not good, as expected, since the distance between points in such color domain does not provide meaningful measurement.

4. CONCLUSION

Using maximum distance clustering (MDC) to generate initial cluster positions for K-means can solve the general problem of clustering-based color reduction methods. It is required to run K-means for only one iteration to prevent it from dominating the process. Thus, the convergence-speed problem of K-means is not present in our algorithm. As MDC is comparatively slow when the number of desired colors is high, a sub-optimal algorithm is proposed and shown to be extremely fast and able to generate a high quality image than many existing interactive color reduction methods, in RGB color space. However, in human perception color space, CIELAB, K-means alone is comparable to MDC + an iteration of K-means, provided that K-means is run for two iterations or higher, one iteration forward and another iteration backward. Considering time, running MDC + an iteration of K-means consumes approximately the same as that of two iterations of K-means.

REFERENCES