APPLICATION OF K-MEANS AND MODIFIED K-MEANS CLUSTERING IN COLOR REDUCTION

Asvini R - EE14BTECH11004, Sanjan Prakash Kumar - EP14BTECH11007

Abstract

Color quantization or color image quantization is a process that reduces the number of distinct colors used in an image, usually with the intention that the new image should be as visually similar as possible to the original image. On systems with 24-bit color displays, true color images can display up to 16,777,216 (i.e., $2^{24} = 256^3$) colors. In this paper, we aim to propose a simple way to implement an algorithm for reducing the number of colors in an image with sufficient visual quality using *K-Means*. Further, we slightly tweak the *classic K-Means* algorithm at the initialization step and compare results [1].

1 Introduction

Color quantization is critical for displaying images with many colors on devices that can only display a limited number of colors, usually due to memory limitations, and enables efficient compression of certain types of images [3]. The infinite number of colors available through the lens of a camera is impossible to display on a computer screen; thus converting any photograph to a digital representation necessarily involves some quantization. Color quantization is mainly used in GIF and PNG images. GIF, for a long time the most popular lossless and animated bitmap format on the World Wide Web, only supports up to 256 colors, necessitating quantization for many images. Some early web browsers constrained images to use a specific palette known as the web colors, leading to severe degradation in quality compared to optimized palettes. The problem of finding optimal palette is computationally intensive, as it is not possible to evaluate all possible color combinations [3]. On the other hand, this can be viewed as a clustering problem to iron out redundancies from the image. This paper implements color reduction using *K-Means clustering* and a slightly *modified version* of it to obtain visually similar images but with lesser colors. Section 2 describes and cites various works related to color quantization and the attempts at it using various clustering techniques. Section 3 lays out the *classic K-Means clustering* and the *modified K-Means clustering* algorithms. Section 4 consists of the results of our endeavors and Section 5 marks the conclusion of our paper.

2 Related Works

Color reduction of images are necessary for several reasons one reason is that the human eye cannot differentiate such a wide range of colors and another reason being that true color image is often too large to serve its purpose in Web applications. Therefore, an effective technique for reducing the number of colors in color images is necessary. The ultimate goal of classical color quantization techniques is to reduce the number of colors of an image with minimum distortion. Therefore, the main objective of computer graphics research in the color quantization area is to select an optimal palette that ensures minimization of a perceived difference between the original and the quantized images [2]. The main issue surrounding image color reduction is to determine which colors are preserved in the resulting image. Using naive strategies such as selecting the most populous colors will result in loss of small and important details on the image. Several techniques have been proposed for color quantization. First, there is a class of techniques that are based on *splitting algorithms*. According to these approaches, the color space is divided into disjoint regions, by consecutively splitting up the color space. In this category belong the methods of median-cut

(MC), and variance-based algorithm. In another major class of color quantization algorithms belong methods based on *cluster analysis*. The clustering techniques frequently used in this category are the Kohonen SOFM, Fuzzy C-means etc. The above techniques are suitable for eliminating the uncommon colors in an image but they are ineffective for image analysis and segmentation [5].

3 K-Means Clustering

K-Means clustering is a vector quantization algorithm that partitions n observations into k clusters. By 'partitions', we mean that the algorithm maps an observation to one of k clusters based on squared (Euclidean) distance of the observation from the cluster centroid. Once the mapping is done, the observation point can simply be represented by the centroid of the cluster to which it is mapped. Applying this on an image would imply reducing it to an image with k colors. One thing to note here is that *K-Means clustering* is an iterative refinement algorithm and the cluster centroids keep getting updated with each iteration. The algorithm terminates when a certain convergence point is reached. The convergence point that is most commonly chosen is when the cluster centroids no longer move [1].

Following are the steps that elaborate on how the *K-Means clustering* algorithm proceeds towards convergence :

1. Start with an initial guess as to what the k cluster centroids could be.

2. For each observation point, find the cluster centroid that is the closest to it in terms of the Euclidean distance. Once found, map it to that particular cluster. This is the *assignment step*.

3. Once all the observations are mapped to some cluster, the new centroids for the next iteration are computed. The mean of all the observations mapped to one particular cluster gives us an updated centroid. This is the *update step*.

4. Steps 2 and 3 constitute a single iteration. They are repeated until the centroids no longer move. This is when we claim 'convergence' to have been achieved.

Initialization : In *K-Means clustering* of any form, we seek to minimize a Euclidean distance between the cluster center and the members of the cluster. The intuition behind this is that the radial distance from the cluster centroid to the element location should 'be similar' for all elements of that cluster. In the steps listed out, the only vague part is regarding the 'initial guess' for the k centroids. This is quite an important part of the algorithm as initializing with an inappropriate set of centroids could lead to convergence to some local minimum, which might be sub-optimal [4].

The *classic K-Means clustering* algorithm selects k observation points randomly from the input. However, with this approach it is quite likely that some of the centroids chosen would be closer to each other than desired. This would defeat the purpose of redundancy reduction in images. This opens up the possibility for an **alternative initialization strategy**. It is an iterative process that builds the initial set of centroids and proceeds as follows :

1. Start off by choosing one of the observation points randomly to be an initial centroid.

2. Assign each observation point to the cluster whose centroid is nearest, using the Euclidean distance as the metric, similar to the assignment step from the *classic K-Means* algorithm. During each assignment, store the Euclidean distance of the observation point from the centroid of the cluster to which it is assigned.

3. Compute new centroids for all the clusters, similar to the update step from the *classic K-Means* algorithm.

4. Using the information from step 2, find the observation point that is farthest away from its centroid and add it to the set of centroids that will form our initial guess.

5. Repeat steps 2 to 4 until the required number of centroids, i.e. k, is obtained.

The intuition behind this approach is that spreading out the k initial cluster centroids is a good thing : the first cluster centroid is chosen uniformly at random from the observation points, after which each subsequent cluster centroid is chosen from the remaining data points with probability proportional to its squared distance from

the point's closest existing cluster centroid. This check is performed in step 4 of the proposed algorithm. This initialization leads to a more assured convergence to an optimal minimum, with the only compromise being the computation time for this initial 'guess'.

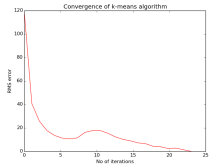
4 Analysis and Results

Figure 1: The image used for analysis. It was a conscious decision to work with an image with a wide contrast spectrum in order to view the results of the color reduction in better light.

The sample image we worked with has a rich palette of colors spanning the extremities of the color spectrum. This was done to ensure that the results of performing clustering on this image would be seen as a well-defined mapping of 'similar' colors to a particular cluster. Thus, the output would be an image with the origin pixels (observation points) replaced by their corresponding cluster centroids. We performed two kinds of clustering algorithms on the sample image which are - *Classic K-Means Clustering* and *Modified K-Means Clustering*. We experimented with different counts of clusters/colors on the same image and also kept track of how the Root Mean Square (RMS) error was varying with the number of iterations. The latter is presented in the form of graphs and can be viewed as a metric for the convergence of the clustering to an appropriate minimum.

Results : The output images and graphs from the *classic K-Means clustering* and the *modified K-Means clustering* have been presented here. These algorithms were run over the input image with the number of clusters being equal to 4, 8, 16 and 32, thus extracting these many significant colors from the original image. This goes a long way in reducing the redundancies in the image by replacing 'similar looking' pixels with one representative (cluster centroid) pixel value.

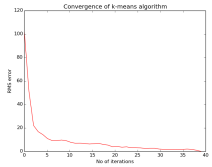
(a) Output image with 4 colors



(b) Convergence graph for 4 clusters

Figure 2: *Classic K-Means* for 4 clusters

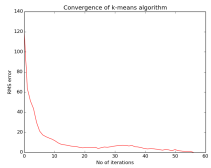
(a) Output image with 8 colors



(b) Convergence graph for 8 clusters

Figure 3: *Classic K-Means* for 8 clusters

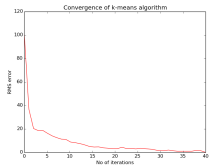
(a) Output image with 16 colors



(b) Convergence graph for 16 clusters

Figure 4: *Classic K-Means* for 16 clusters

(a) Output image with 32 colors

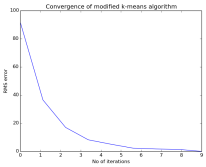


(b) Convergence graph for 32 clusters

Figure 5: *Classic K-Means* for 32 clusters

The *classic K-Means clustering* algorithm picks k random observation points as the initial guess for the k cluster centroids. This approach runs the risk of selecting the initial k cluster centroids to be closer to each other than desired, which could lead to the convergence towards a sub-optimal minimum. This point shall be highlighted further when discussing the results from the *modified K-Means clustering* algorithm.

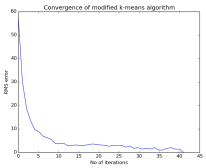
(a) Output image with 4 colors



(b) Convergence graph for 4 clusters

Figure 6: *Modified K-Means* for 4 clusters

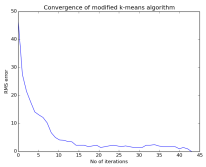
(a) Output image with 8 colors



(b) Convergence graph for 8 clusters

Figure 7: *Modified K-Means* for 8 clusters

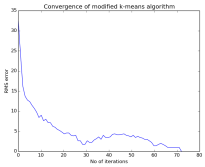
(a) Output image with 16 colors



(b) Convergence graph for 16 clusters

Figure 8: *Modified K-Means* for 16 clusters

(a) Output image with 32 colors



(b) Convergence graph for 32 clusters

Figure 9: *Modified K-Means* for 32 clusters

The *modified K-Means clustering* algorithm ensures that the initial guess for the k cluster centroids are well spread out, thus facilitating a more optimal elimination of redundancies in the input image. On comparing results, we see that the image obtained from the *modified K-Means clustering* algorithm is in greater resemblance with the input image than that obtained from the *classic K-Means clustering* algorithm. This statement is brought to light mainly by the outputs with smaller number of clusters, in our case being 4. As the number of clusters increase, the output image becomes finer in detail and seems to be beyond our capabilities of perception to compare similarities with the original image, which is why the outputs of the *classic K-Means clustering* algorithm and the *modified K-Means clustering* algorithm look alike. Further, with the *modified K-Means clustering* algorithm, we also notice a smoother convergence to a minimum than that seen in the *classic K-Means clustering* algorithm as illustrated by the graphs displayed.

5 Conclusion

The *K-Means clustering* algorithm is a useful tool in the color quantization domain, as illustrated by our paper, in practice. The *modified K-Means clustering* algorithm led to a visible improvement over the *classic K-Means clustering* algorithm for smaller number of clusters. However, for larger number of clusters, like 16 or 32, this improvement was modest and perhaps scarcely visible. On the flip side, it does guarantee a better spread of clusters than the *classic K-Means clustering* algorithm. This method of initialization could be employed in a pre-processing stage before much advanced image processing and computer vision algorithms from the industry take over.

6 Bibliography

[1] Tom Mikolov (2008) : ”Color Reduction Using K-Means Clustering”, Faculty of Information Technology, Brno University of Technology, Brno (Czech Republic)

[2] Joo Hyun Song (2010) : ”Effective Color Reduction Using the Modified Diversity Algorithm”, University of Iowa (USA)

[3] Michael T Orchard, Charles A Bouman (1991) : ”Color Quantization of Images”, IEEE

[4] Miti Ruchanurucks (2007) : ”Clustering-based Color Reduction”, University of Tokyo (Japan)

[5] Nikos Papamarkos, Antonis E Atsalakis, Charalampos P Strouthopoulos (2002) : ”Adaptive Color Reduction”, IEEE

7 Appendix

Sample pictures for reduction to 16 colors/clusters :

(a) Original image

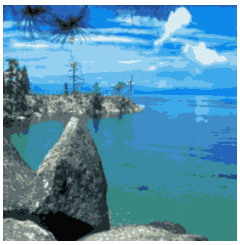
(b) K-means clustering for 16 clusters

(c) Modified K-means clustering for 16 clusters

Figure 10: SAMPLE IMAGE I

(a) Original image

(b) K-means clustering for 16 clusters



(c) Modified K-means clustering for 16 clusters

Figure 11: SAMPLE IMAGE II

(a) Original image

(b) K-means clustering for 16 clusters

(c) Modified K-means clustering for 16 clusters

Figure 12: SAMPLE IMAGE III