

Image compression via K-means and SLIC superpixel approaches

Xiaolin Luo^{1,2,a}

¹Chongqing Zhongyu Engineering Consultation & Supervision Co., Ltd, Chongqing, 400065, China

²Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

^aemail: 410314755@qq.com

Keywords: Image Compression; K-means; Superpixel

Abstract. Image compression is a key component in the transmission and storage process of image data. The major objective of image compression is to reduce the irrelevance and redundancy of the image data in order to store and transmit them in an efficient form. One of the practical approaches is to reduce the original large color space to a considerable small scale. In detail, we implement this image compression method via two approaches: K-means, which directly clustering the colors to nearest centroids, and the SLIC superpixel approaches, which relies on generating computationally efficient and perceptually meaningful image segments. Experiment results reveal that both of our approaches could effectively compress the image size.

Introduction

Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from web pages. There are several different ways in which image files can be compressed. For Internet use, the two most common compressed graphic image formats are the JPEG format and the PNG [1] format. The JPEG method is more often used for photographs, while the PNG method is commonly used for line art and other images in which geometric shapes are relatively simple. Other techniques for image compression also include the use of fractals and wavelets [2]. Nevertheless, these methods have not gained widespread acceptance for use on the Internet as of this writing.

On the other hand, these above mentioned methods generally assume that the image qualities should be greatly preserved. From this perspective, the computationally complexities of these kind of algorithms may increase due to the reason that the most representative feature components need to be automatically and adaptively selected for each individual image. However, in some cases, the image quality may not be the most crucial consideration in the whole system, i.e., we do not need the best qualities of the images during transmission, such occasions include image retrieval via thumbnails [3]. Hence, here comes a question, how to efficiently compress images further with tolerable quality loss?

In this paper, we address this problem via reducing the number of colors contained in an image directly. More specifically, two different methods: K-means clustering [4] and SLIC superpixel [5] are employed independently to replace each pixel in the image with its nearest cluster centroid color. The K-means algorithm is directly conducted over a small scale image of the original one in order to obtain the quantized color centroids efficiently. The K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. It aims to partition n observations into k clusters in which each observation belongs to the cluster with nearest mean, serving as a prototype of the cluster. Later, each individual pixels in the original large image are classified to the nearest quantized color centroids. On the other hand, the SLIC superpixel first divides the small scale image into grids, and later the color centroids are generated via K-means conducted in these grids. In experiments, we find that both approaches could effectively compress the image size, whereas the K-means superpixel could better preserve the minute textures and details; and SLIC could earn better compression ratio.

Methodology

In general, our image compression framework consists of two stages: color centroids generation within small scale image I_{small} and color re-computing in the original image I , which is illustrated in Fig. 1. In order to reduce the image color space from the large scale n down to a small scale k , we first conduct the color centroids generation procedures on a resized small image I_{small} of the original one I . The motivation of employing I_{small} rather than I to compute color centroids is the color centroids are not sensitive to the image size with K-means and superpixel algorithms, which will be discussed later. In addition, one of our major assumption is that we could have tolerable image loss or distortion. On the other hand, the optimization process of the K-means or superpixel algorithms is NP-hard, hence, the Expectation Maximization (EM) algorithm [6] is employed to obtain the optimum color centroids, whereas the process is of great time complexity. After we have pre-processed the given image to resized size and obtained the corresponding image centroids, we then use the generated k colors to render the original image I , i.e., each pixel $I(x,y)$ in the original image are assigned with new value from one of the k color centroids. Finally, we could obtain the compressed image I_{comp} as the image compression result.

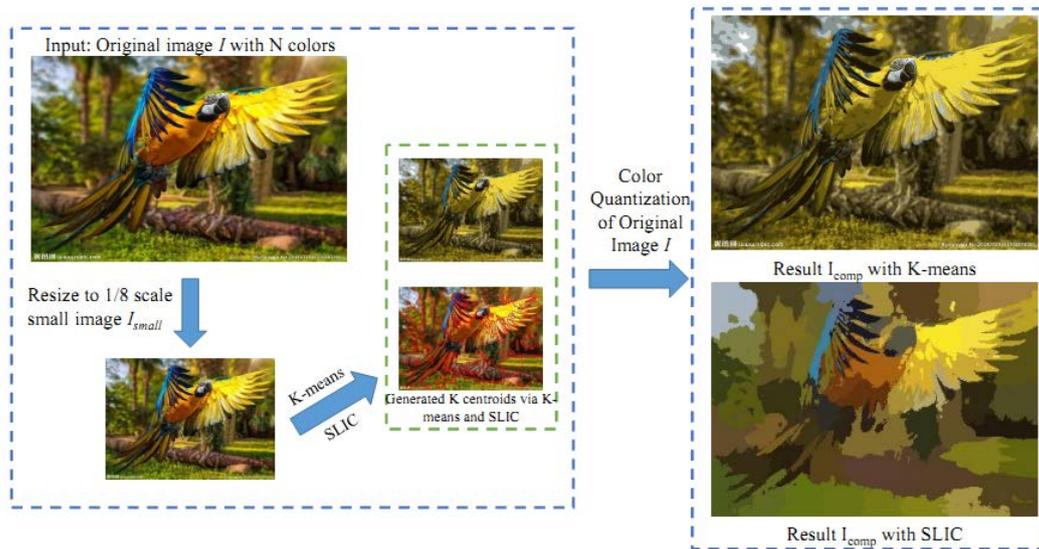


Fig.1. The general pipeline of our proposed image compression method

Centroids Generation using K-means

In the color centroids generation problem with K-means, we are given all the image pixels in the resized image I_{small} as the training set $\{I(1), I(2), I(3), \dots, I(h*w)\}$, where h and w are the resized image's height and width. Our goal is to group these image pixels into a k cohesive *clusters*, and the centers of the *clusters* are exactly the k color centroids we want. Because there are no labels $y(i)$ given, this is an unsupervised learning problem. From the numerical optimization perspective, the K-means problem could be viewed as minimizing the following objective function:

$$J(c, \mu) = \sum_{i=1}^{h*w} \|I(i) - \mu_c(i)\|^2 \quad (1)$$

In the type: J measures the sum of squared distances between each training example $I(i)$ and the cluster centroid $\mu_c(i)$ to which it has been assigned. Hence, one fact in the K-means optimization problem is that two variables c and μ need to be optimized simultaneous, and the c could be further viewed as a latent variable. In general, these kind of problem could not be optimized directly with closed form solution.

However, it can be shown that K-means is exactly coordinate descent on J , with an alternative optimization strategy: EM algorithm being employed. Specifically, the inner-loop of K-means repeatedly minimizes J with respect to c while holding μ fixed (E-step), and then minimizes J with respect to μ while holding c fixed (M-step). Thus, J must monotonically decrease, and the value of J must converge.

Centroids Generation using SLIC

A superpixel can be defined as a group of pixels which have similar characteristics. It is

generally color based segmentation. Superpixels can be very helpful for image segmentation. Compared with color quantization methods based on individual pixels, the superpixel approaches could provide a convenient primitive to preserve local image features. In this paper, we implement the superpixel on the basis of *simple linear iterative clustering* (SLIC) algorithm. In spite of directly employing RGB color space in global I_{small} to conduct K-means color centroid generation, the SLIC algorithm first divides the I_{small} into K grids, with a grid interval $S = \sqrt{h*w/(K*K)}$. Later, the SLIC algorithm performs a local clustering of pixels in 5-D space defined by the L, a, b values of the CIELAB color space along with the x, y pixel coordinate. In this degree, the k cluster centers could be represented with a vector: $C_k = [l_k, a_k, b_k, x_k, y_k]$, with $k = [1, K]$ at regular grid intervals S .

Instead of employing Euclidean norm measurement in our previous K-means centroids generation procedures, SLIC proposes a new distance measure D_s , which is described as below:

$$\begin{aligned} d_{lab} &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{xy} &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ D_s &= d_{lab} + \frac{m}{S} d_{xy} \end{aligned} \quad (2)$$

In this type: m is designed to allow us control the compactness of the superpixel, the greater of the value m , the more spatial compact the cluster is. In addition to the new distance measurement of the $Labxy$ space, the SLIC algorithm also utilizes a gradient based sampling centroids disturbance strategy on the initialization of the centroids. In other words, the algorithm begins by sampling K regularly spaced cluster centers and moving them to the seed locations corresponding to the lowest gradient position in a $3*3$ neighborhood. Empirical experiments reveal that this strategy could greatly reduce the chances of choosing a noise pixel when comparing with purely randomly sampling in K-means. Finally, the SLIC runs clustering in each grids, the whole pipeline is as follow: each pixel in the I_{small} is associated with the nearest cluster center in the searching area. After all pixels are associated to the present stage centroids C_k , the new centers are updated as the average $Labxy$ vector of all assigned pixels. This process is iteratively repeated until convergent. At last, a connectivity enforcement process is conducted by relabeling disjoint segments with the labels of largest neighboring cluster.

Experiments and Results

In the experiment section, we evaluate two different image compression methods, respectively from the following perspectives: image compression results; image compression ratio and computational complexity. 5 different object images are selected named as: bird, car, dog, flower, and dog. The images' size ranging from dozens of pixels to thousands, whereas the storage sizes ranging around hundreds KB. We use *matlab* to implement the two algorithms. In the following experiments, we set the number of color centroids at 16 for both K-means and SLIC algorithm. The I_{small} is generated via resizing the original image to 1/8 scale. In addition, both of the algorithms are iterated for 100 rounds at most, and will be terminated if less than 3% relatively decreasing in objective function is found. Finally, the overall detailed information and compression results are given in Fig.2 and Table.1.

From Fig.2 and Table.1, it is clear that both of our algorithm successfully compressed the original images I to those with relatively smaller ones. In particular, according to Table.1, the compression with K-mean color centroids generation strategy compresses 5 images with a compression ratio at 64%. On the other hand, the SLIC superpixel counterpart achieves over 98% compression ratio. On the other hand, we find that SLIC algorithm takes more computation time than the K-means method.

Another interesting phenomenon we found is the images' style after conducting the compression procedures are totally different. From Fig.2, we can find that compressed images using K-means generally has a sharpening effect, i.e., most of the textures are preserved whereas the global image style seems like the oil painting. On the other hand, the SLIC algorithms compresses the images more aggressively: the whole image are divided into several semantic segments, whereas most of

the details are lost. This is also coherent with the differences in the compression ratio using different algorithms, as illustrated in Table.1. We may conclude that the K-means may be more suitable to those occasions need better textures and detail information and vice versa.

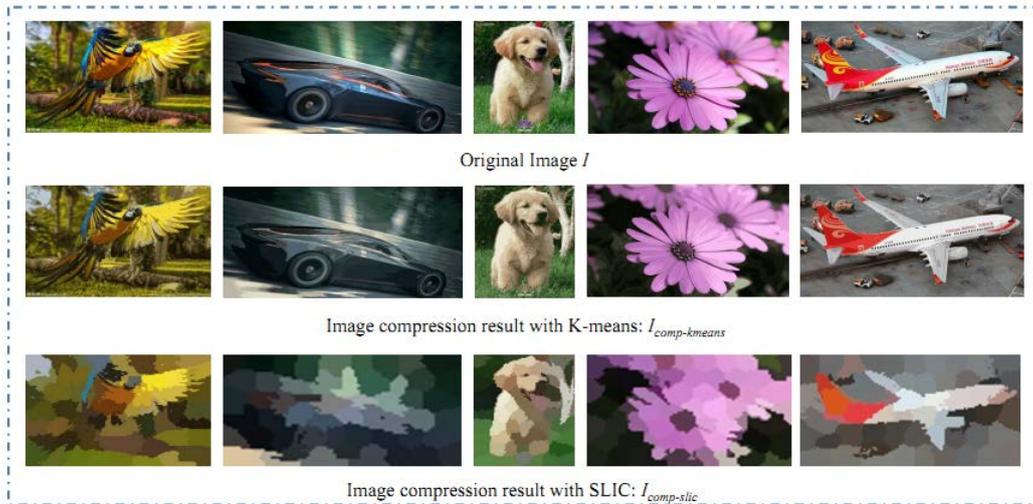


Fig.2. Row 1: The original images; Row 2: Results using K-means; Row 3: Results with SLIC superpixel.

Table.1 The image info and numerical compression results

Image Info	Compression with K-means		Compression with SLIC superpixel	
	Compressed size	Computing time	Compressed size	Computing time
Bird (1024*740 588KB)	133.2KB	146.3s	35.6KB	157.3s
Car (2560*1440 840KB)	318.2KB	182.2s	32.5KB	202.1s
Dog (600*800 166KB)	91.3KB	110.7s	23.4KB	132.4s
Flower (631*402 44KB)	34.0KB	94.1s	16.0KB	115.6s
Plane (1000*500 80.6KB)	52.3KB	123.5s	22.0KB	142.9s
Average (342.72KB)	125.8KB (64%)	131.34s	25.9KB (98%)	150.1s

Conclusions

In this paper, we address the image compression topic with the assumption that tolerable image quality loss or distortion is permitted. We then compress the image by reducing the number of colors from N to K . In order to realize this goal, we first resize the original image to a smaller one; later, two common algorithms K-means and SLIC superpixel are employed respectively to calculate the color centroids of the small scale image; finally, the pixels in the original images are quantized to the nearest color centroids based on different distance measurements. Experiment results reveal that both of our algorithms could reduce the storage size of the original image dramatically, whereas the K-means approaches could better preserve the detailed local textures with higher computational complexity. On the other hand, SLIC approach can achieve higher compression ratio, with more loss in detail information.

References

- [1] David Taubman, Michael Marcellin. JPEG2000 Image compression Fundamentals, Standards and Practice [J], 2012.
- [2] Yuval Fisher. Fractal image compression [J]. Fractals, 1994 2(3) 347-361.

- [3] Thomas Deselaers, Daniel Keysers, Hermann Ney. Features for image retrieval: an experimental comparison [J]. *Information Retrieval*, 2008 11(2) 77-107.
- [4] Alon Vinnikov, Shai Shalev-Shwartz. K-means recovers ICA filters when independent components are sparse. *International Conference on Machine Learning [C]*. Berlin: Springer, 2014.1234-1237.
- [5] Radhakrishna Achanta, Appu Shaji, Kevin Smith, et al. SLIC superpixels compared to state-of-the-art superpixel methods [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012 34(11) 2274-2282.
- [6] Yin Jiangtao, Zhang Yanfeng, Gao Lixin. Accelerating Expectation-Maximization algorithm with frequent updates. *IEEE International Conference on Cluster Computing [C]*. New York: IEEE, 2012.537-540.