

Research of Computer Desktop Image Compression Clustering Algorithm

Ronghua Qiao

School of Information Engineering, Yulin University, Yulin, Shanxi, China, 719000
 liuxiangjuaa22@163.com

The basis for the image algorithm is the Discrete Cosine Transform (DCT) which extracts spatial frequency information from the spatial amplitude samples. In this paper, an image compression k-means clustering algorithm based on block-based discrete cosine transform to compress computer desktop image is proposed. It divides computer screen image into 16×16 non-overlapping blocks, then each block is classified into text/graphic block, hybrid block or picture block. For text/graphic blocks with rich color, a color clustering algorithm is used to reduce the number of colors, text/graphic block is coded by lossless compression and hybrid block is coded by hybrid coding method. Experimental results indicated that the image by our proposed algorithm has higher Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) than traditional algorithms.

1. Introduction

Data compression is the reduction or elimination of redundancy in data representation in order to achieve savings in storage and communication costs. The objective of an image compression technique is to represent an image with smaller number of bits without introducing appreciable degradation of visual quality of decompressed image. In a digital true color image, each color component that is R, G, B components, each contains 8 bits data (Donnell et al., 2011). Most color images are recorded in RGB model. However, RGB model is not suited for image processing purpose. Therefore, RGB images are transformed to one of the luminance-chrominance models, performing the compression process, and then transform back to RGB model because displays are most often provided output image with direct RGB model.

A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical for compression, since it turns out that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions.

Douak proposed a new algorithm for color images compression (Douak et al., 2011). After a preprocessing step, the DCT transform is applied and followed by an iterative phase including the threshold, the quantization, dequantization and the inverse DCT. For obtaining the best possible compression ratio, the next step is the application of a proposed adaptive scanning providing, for each (n, n) DCT block a corresponding (n×n) vector containing the maximum possible run of zeros at its end.

Zhao proposed a hybrid image compression method, the background of the image is compressed using lossy compression and the rest of the image is compressed using lossless compression. In hybrid compression of color images with larger trivial background by histogram segmentation, input color image is subjected to binary segmentation using histogram to detect the background. The color image is compressed by standard lossy compression method. The difference between the lossy image and the original image is computed and is called as residue. The residue at the background area is dropped and rest of the area is compressed by standard lossless compression method. This method gives lower bit rate than the lossless compression methods and is well suited to any color image with larger trivial background (Zhao et al., 1996).

In this paper, we proposed a color image compression method, which makes a balance on compression ratio and image quality by compressing the vital parts of the image with high quality. In our approach, the main subject in the image is very important than the background image. Considering the importance of image components and the effect of smoothness in image compression, the method use clustering algorithm to classify the image as edge blocks (main subject) and nonedge blocks (background), then the background of the image is subjected to low quality lossy compression and the main subject is compressed with high quality lossy compression.

The rest of the paper is organized as follow, The JPEG Compression method and clustering algorithm used in the proposed work are described in section 2. Section 3 describes the proposed algorithm. Section 4 presents the experimental results obtained in the paper. Section 5 draws the conclusion of this work.

2. Background

2.1 JPEG image compression technique

JPEG is a commonly used method of lossy compression for digital images. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. The JPEG standard specifies the codec, which defines how an image is compressed into a stream of bytes and decompressed back into an image, but not the file format used to contain that stream.

JPEG uses a lossy form of compression based on the discrete cosine transform (DCT). There are three closely connected processes namely, encoding, quantization and decoding for JPEG compression.

In the image compression algorithm, the input image is divided into 8×8 or 16×16 non-overlapping blocks, and the two-dimensional DCT is computed for each block. The DCT coefficients are then quantized, coded, and transmitted. The JPEG receiver decodes the quantized DCT coefficients, computes the inverse two-dimensional DCT of each block, and then puts the blocks back together into a single image.

The architecture of the JPEG encoder is shown in Figure 1.

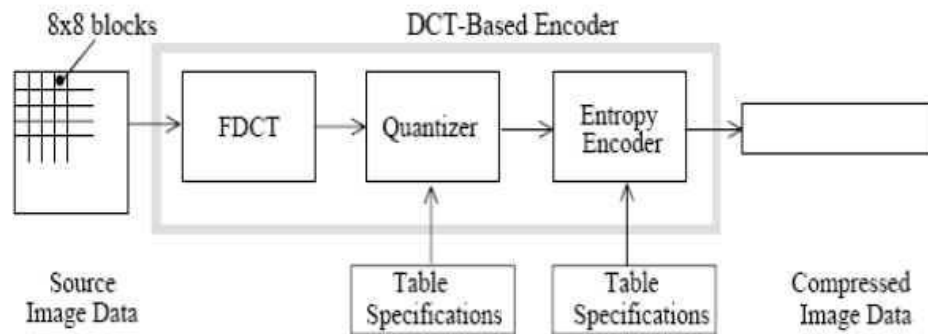


Figure 1: Block diagram of JPEG encoder

The encoding process consists of several steps:

1. The representation of the colors in the image is converted from RGB to $Y'CbCr$, consisting of one luma component (Y'), representing brightness, and two chroma components, (C_B and C_R), representing color.
2. The resolution of the chroma data is reduced, usually by a factor of 2 or 3. This reflects the fact that the eye is less sensitive to fine color details than to fine brightness details.
3. The image is split into blocks of 8×8 pixels, and for each block, each of the Y , C_B , and C_R data undergoes the discrete cosine transform (DCT). A DCT is similar to a Fourier transform in the sense that it produces a kind of spatial frequency spectrum.
4. The amplitudes of the frequency components are quantized. Human vision is much more sensitive to small variations in color or brightness over large areas than to the strength of high-frequency brightness variations. Therefore, the magnitudes of the high-frequency components are stored with a lower accuracy than the low-frequency components. The quality setting of the encoder affects to what extent the resolution of each frequency component is reduced. If an excessively low quality setting is used, the high-frequency components are discarded altogether.
5. The resulting data for all 8×8 blocks is further compressed with a lossless algorithm, a variant of Huffman encoding.

The decoding process reverses these steps, except the quantization because it is irreversible. The architecture of the JPEG decoder is shown in Figure 2.

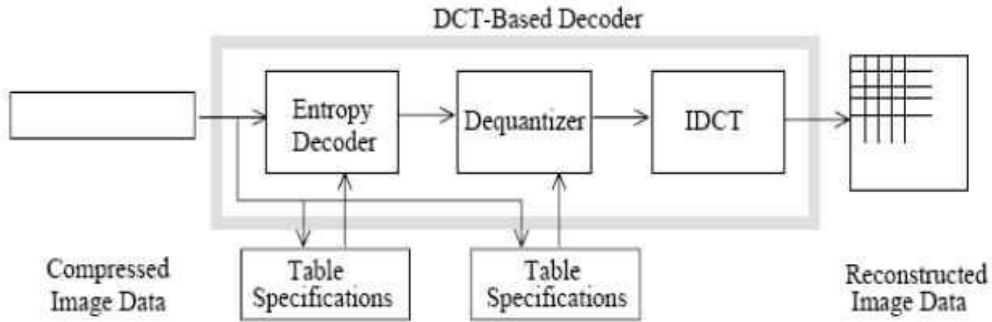


Figure 2: Block diagram of JPEG decoder

As an example, one such 8×8 8-bit image might be:

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Before computing the DCT of the 8×8 block, its values are shifted from a positive range to one centered around zero. For an 8-bit image, each entry in the original block falls in the range [0, 255]. The midpoint of the range is subtracted from each entry to produce a data range that is centered around zero, so that the modified range is [-128, 127]. This step results in the following values:

$$g = \begin{matrix} \xrightarrow{x} \\ \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} \\ \downarrow y \end{matrix}$$

The next step is to take the two-dimensional DCT, which is given as Eq(1):

$$G_{u,v} = \frac{1}{4} \alpha(u)\alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right] \tag{1}$$

Where, u is the horizontal spatial frequency, v is the vertical spatial frequency, $\alpha(u)$ is a normalizing scale factor to make the transformation orthonormal.

For decoding purpose there is an inverse DCT (IDCT), which is given as Eq(2):

$$g_{x,y} = \frac{1}{4} \sum_{x=0}^7 \sum_{y=0}^7 \alpha(u)\alpha(v) G_{u,v} \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right] \tag{2}$$

After output from the forward DCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a carefully designed 64–element quantization table. At the decoder, the quantized values are multiplied by the corresponding quantization table elements to recover the original unquantized values. After quantization, all of the quantized coefficients are ordered into the “zigzag” sequence as shown in Figure 3.

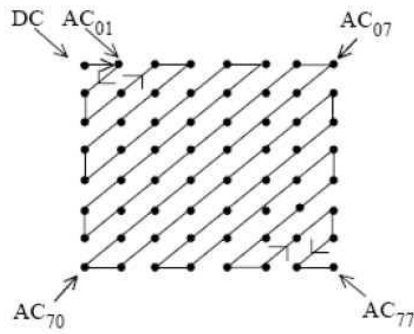


Figure 3: Zigzag Sequence

2.2 k-means clustering

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

The algorithm follows a simple and easy way to classify a given data set through a certain number of clusters (k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early grouping is done. At this point we need to re-calculate k new centroids as mean of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done (Kanungo et al., 2002). In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, which is defined as Eq(3):

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (3)$$

where $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j is an indicator of the distance of the n data points from their respective cluster centres (Hartigan et al., 2013).

3. Computer desktop image compression clustering algorithm

In order to effectively compress the computer desktop image, this paper proposes a new compression algorithm based on the different characteristics of blocks. The algorithm divides computer screen image into 16×16 non-overlapping blocks, then each block is classified into text/graphic block, hybrid block or picture block. Lossless compression coding method is used for text/graphic block. Lossy compression coding method, similar to H.264 intra prediction, is used for picture block. Our proposed hybrid compression coding method is used for hybrid block.

The proposed hybrid compression coding method built around several steps. Each step will be explained in more details in the following:

Firstly, before the application of the RGB to $YCbCr$ transformation, the mean values of the three plane images R, G and B are removed. Consequently, we can achieve an efficient compression that allows reaching high compression ratios in the C_b and C_r without losing the quality of the whole compressed image when returned to the original RGB space. It means that acting on $YCbCr$ space proffers better performances than the original RGB space.

Secondly, for any color image, after the preprocessing step (means removing and RGB to $YCbCr$), each one of the new three planes (Y , C_b , C_r) are partitioned to blocks. k-means clustering algorithm is used for edge detection. It will classify the blocks into edge (foreground and more important block) and non-edge (background and less important block).

At last, each block is DCT transformed. It is clear, that DCT transform concentrate the great part of block energy in few representative coefficients. The DCT coefficients in each block are then uniformly quantized with quantization step sizes depending on the DCT coefficient. The step sizes are represented in a quantization matrix called the Q-matrix. Different Qmatrices are typically used for the luminance and chrominance components. This quantization stage determines both the amount of compression and the quality of the

decompressed image. Large quantization step sizes give good compression but poor visual performance while small quantization step sizes give good visual performance but small compression.

4. Experiment and result analysis

In order to test the compression performance of our proposed algorithm for the computer desktop image, two computer desktop images of 1024 x 768 pixels were selected in this paper. The two computer desktop images are shown in Figure 4.

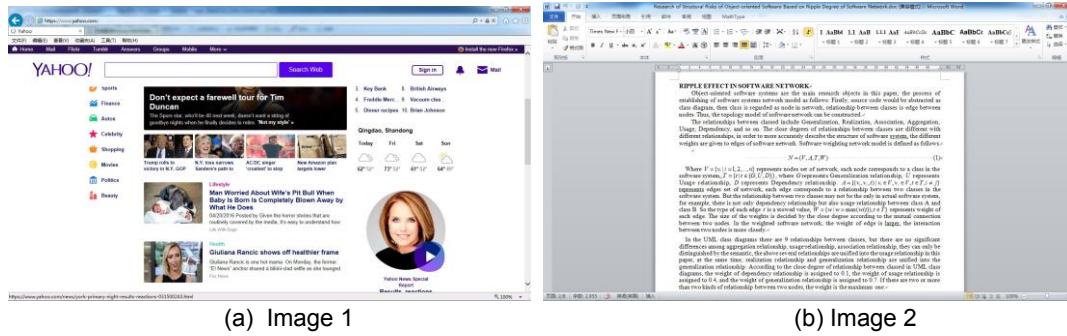


Figure 4: Original test images

In order to verify our proposed algorithm in terms of the improvement of computer desktop image compression, other 3 compression algorithms (JPEG, JPEG2000, JPEG-LS) (Taubman et al., 2002) are selected and compared. Comparing results of PSNR and SSIM are individually shown in Figure 5 and Figure 6.

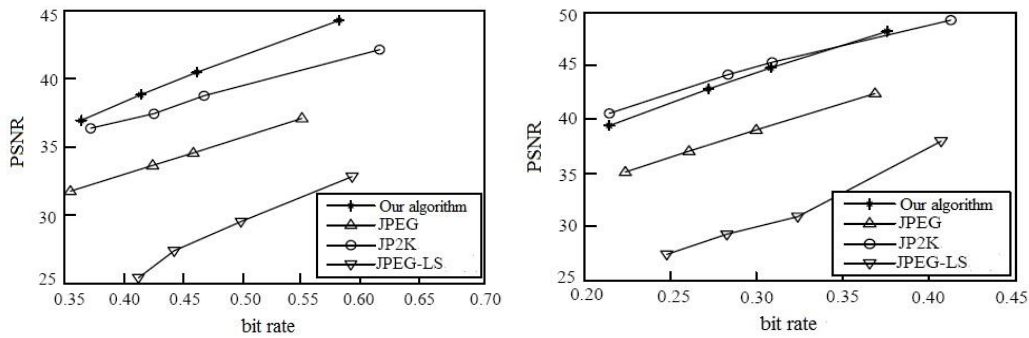


Figure 5: Image1 and Image2 PSNR comparison of different algorithms

From Figure 5 we can see, our proposed algorithm is better than JPEG and JPEG-LS compression algorithm for each bit rate. In most cases, the compression performance of our algorithm is better than that of JPEG2000 algorithm.

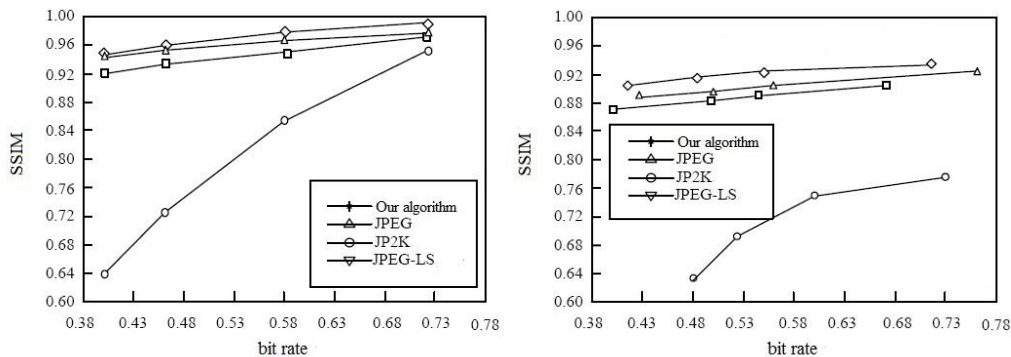


Figure 6: Image1 and Image2 SSIM comparison of different algorithms

From Figure 6 we can see, the SSIM value of our algorithm is higher than that of JPEG, JPEG2000 and JPEG-LS algorithm.

5. Conclusions

In this paper, a computer desktop image compression algorithm based on color clustering is proposed. The algorithm divides computer screen image into 16×16 non-overlapping blocks, then each block is classified into text/graphic block, hybrid block or picture block. Then different methods are used to compress different blocks. Experimental results show that our proposed algorithm can correctly distinguish different block categories and improve the compression efficiency.

Acknowledgment

This paper is supported by Science and Technology plan projects of Yulin, China (No. 2015CXY-20-1)" research and development of key technologies for the integration of "Internet +" information platform", and is supported by youth foundation of Yulin University (No.14YK21) "research on Relevance feedback in content-based image retrieval".

Reference

- Donnell O., Goynes W., 2011, Determination of representative elementary areas for soil redoximorphic features identified by digital image processing, *Geoderma*, 3, 138-146, DOI: 10.1016/j.geoderma.2010.12.011
- Douak F., Benzid R., 2011, Color image compression algorithm based on the DCT transform combined to an adaptive block scanning, *International Journal of Electronics and Communications*, 65, 16-26, DOI: 10.1016/j.aeue.2010.03.003
- Hartigan J., Wong M., 1979, A K-means clustering algorithm, *Applied Statistics*, 1, 100-108, DOI: 10.2307/2346830
- Kanungo T., Mount D. M., Netanyahu S., 2002, An Efficient k-Means Clustering Algorithm: Analysis and Implementation, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 7, 881-892, DOI: 10.1109/TPAMI.2002.1017616
- Taubman, S., Marcellin W., 2002, JPEG2000 Image Compression Fundamentals, Standards and Practice, Springer International, 2, 286- 288, DOI: 10.1007/978-1-4615-0799-4
- Zhao Y., Yuan B., 1996, A hybrid image compression scheme combining block-based fractal coding and DCT, *Signal Processing Image Communication*, 2, 73-78, DOI: 10.1016/0923-5965(95)00036-4