Blank Background Image lossless Compression Technique

Samer Sawalha *samerfs@hotmail.com*
Information and Communication Technology
Royal Scientific Society
Amman-Jordan

Arafat Awajan *awajan@psut.edu.jo*
King Hussein School for Computing Sciences
Princess Sumaya University for Technology
Amman-Jordan

Abstract

This paper presents a new technique able to provide a very good compression ratio in preserving the quality of the important components of the image called main objects. It focuses on applications where the image is of large size and consists of an object or a set of objects on background such as identity photos. In these applications, the background of the objects is in general uniform and represents insignificant information for the application. The results of this new techniques show that is able to achieve an average compression ratio of 29% without any degradation of the quality of objects detected in the images. These results are better than the results obtained by the lossless techniques such as JPEG and TIF techniques.

Keywords: Image Compression, Blank Background Images, Histogram Based Thresholding, Background Detection, Object Detection.

1. INTRODUCTION

Storing huge amount of high quality images may cause storage capacity problems and speed down their access. Image compression is the process of representing an image with fewer bits while maintaining image quality, saving cost associated with sending less data over communication lines and finally reducing the probability of transmission errors [5].

There are many previous studies and techniques which have dealt with compressing images. These techniques are grouped into two classes: the lossy compression techniques and the lossless compression techniques. The lossless compression algorithms save the quality and information but their performance in term of compression ratio and time access are weak. The lossy compression algorithms destroy the quality, crop the image, change its dimensions or remove some important information from it.

Applications such as Civil Affairs Department information system (CAD) and Human Resources information systems manipulate huge amount of pictorial information. These applications store the images either in separate files or in databases using Oracle, MySQL, SQL Server. These applications face problems of storage capacity and access time to these images due to the huge amount of high quality images sizes [1] [2] [3]. The main objective of this work is to develop a technique to store large sized images. This technique should preserve the original dimensions of the image and its resolution, and decrease its size. We are focusing on applications where the image background does not carry important information since the critical information is only the object or objects on the background of the image. The face component in an identity photos in CAD application is an example of these application.

2. PREVIOUS STUDIES

During the past decades, a lot of studies were conducted on image compression. These researches succeeded to provide good number of compression techniques able to reduce the size of the images. Each one of these works has its own objectives, advantages and disadvantages. In this section, we will discuss some of these compressions techniques [4]. The image compression techniques are broadly classified into two categories depending on whether if we can restore the exact replica of the original or not. These categories are the lossless image compression techniques and the lossy image compression techniques [6].

2.1 Lossless Image Compression

Lossless image compression techniques allow the image to be encoded (compressed) into a smaller size without losing any details of the image. The compressed image can then be decoded in order to retrieve the original image [6]. The main techniques belonging to this category are RLE, LZW and Huffman encoding.

2.1.1 Run Length Encoding (RLE)

It is a simple and direct compression technique. It takes advantage of the fact that lot of adjacent pixels are often identical in the image [3]. RLE checks the stream of pixels and inserts a special token each time a chain of more than two equal adjacent pixels value are found. This special input advises the decoder to insert the following pixel value n times into its output stream [7]. RLE can be easily implemented, either in software or in hardware. It is fast and very well verifiable However, its compression performances are very limited mainly when the lighting of the image produce makes some kind of gradient.

2.1.2 Lempel-Ziv-Welch (LZW) Coding

It assigns fixed length code words to variable length sequences of coding symbols without a priori knowledge of the symbols probabilities. At the onset of the coding process, a dictionary containing the source symbols to be coded is constructed, for 8-bit monochrome images, the first 256 words of the dictionary are assigned to intensities 0, 1,255 [1]. As the encoder sequentially examines image pixels, intensity sequences that are not in the dictionary are placed in determined locations. For instance, if the first two pixels are white, sequence “255-255” might be assigned to 256. Next time – when two consecutive white pixels are encountered – code word 256 will be used to represent them [1].

2.1.3 Huffman Encoding

This technique is based on the statistical occurrence of symbols. In image compression, the pixels in the image are treated as symbols. The symbols that occur more frequently are assigned a smaller number of bits, while the symbols that occur less frequently are assigned a relatively larger number of bits. Huffman code is a prefix code. This means that the binary code of any symbol is not the prefix of the code of any other symbol. Most image coding standards use lossy techniques in the earlier stages of compression and use Huffman coding as the final step [3].

2.2 Lossy Image Compression

The lossy image compression methods are better in compression ratio than the lossless methods. However, the generated image will not be same as original image: some of information may disappear and the quality may decrease [6]. These techniques are widely used in application where the size of the image and speed of transmission is more critical than the quality of the image. There are many examples of lossy image compression methods such as JPEG [8], JPEG2000 [8] and Transformation coding [3].

2.2.1 JPEG

Is created by the Joint Photographic Experts Group in 1983 and adopted by both the International Standards Organization (ISO) and International Telegraph Union Telecommunications standards sector (ITU-T). It is the most commonly used method for compressing digital images.

JPEG reduces the size of an image by breaking it into 8x8 blocks and within each block, shifting and simplifying the colours so there is less information to encode. The pixels are changed only in relation to the other pixels within their block, two identical pixels that are next to each other, but in different blocks, could be transformed into different values. When high compression ratios are used, the block boundaries become obvious, causing the 'blockiness' or 'blocking artifacts' frequently observed in common JPEG [8].

2.2.2 JPEG2000

It uses state-of-the-art compression techniques based on wavelet technology. The JPEG2000 standard was adopted by ISO. It is a wavelet-based compression algorithm that offers superior compression performance over JPEG at moderate compression ratios. The end result is a much better image quality however at high levels of compression the image appears blurred [8].

2.2.3 Transformation Coding

In this coding scheme, transformations such as Discrete Fourier Transform DFT and Discrete Cosine Transform DCT are used to transform the pixels in the original image into frequency domain coefficients. These coefficients have several desirable properties such as the energy compaction property. The results are concentrated in only a few of the significant transform coefficients. This is the basis of achieving the compression. Only those few significant coefficients are selected and the remaining is discarded. The selected coefficients are considered for further quantization and entropy encoding. DCT coding has been the most common used approach; it is also adopted in the JPEG image compression standard [3].

3. IMAGE COMPRESSION

Our proposed technique consists of a sequence of operations aiming at determining the background of the image, extracting identifying objects and compressing the image. These operations can be described by the following algorithm:

a) Transform the image into Grayscale image.
b) Histogram generation
c) Threshold calculation
d) Average background colour selection
e) Search for an object in the image
f) Object extraction
g) Image object compression
h) Object subtraction
i) Go to 5 until no objects are found
j) Store extracted objects and background colour

3.1 Grayscale Image Generation

The manipulated images are in general coloured image where each pixel is represented by three values representing the three main components or colours: Red, green and blue. The proposed technique starts by transforming the colour image into grayscale image. A grayscale image is an image where each pixel has a single value representing the intensity or brightness information at that pixel. The range of each pixel value is between 0 and 255 if we associate one byte for each pixel.

The luminosity method which is a more sophisticated version of the average method is used in this work to generate the grayscale image. It calculates a weighted average of luminosity according to the formula (0.21 Red value + 0.71 Green value + 0.07 Blue value).

3.2 Background Extraction

In many applications, the gray levels of pixels belonging to the object are generally different from the gray levels of the pixels belonging to the background. This fact is a general features in applications such as identity photos in CAD applications. Thus, thresholding becomes a simple

but effective tool to separate objects from the background [9]. The selection of a threshold value is then a critical operation preceding the extraction of the background.

Image thresholding is one of the most important techniques for image segmentation aiming at partitioning an image into homogeneous regions, background and main objects in our case. It is regarded as an analytic image presentation method and plays a very important role in many tasks of pattern recognition, computer vision, and image and video retrieval [1].

There are two main groups of thresholding algorithms: Fixed threshold and histogram derived threshold. In the fixed threshold selection, the threshold is chosen independently of the image data. In the histogram derived threshold selection, the threshold is selected from the brightness histogram of the region or image that we wish to segment. Since the images are different in colour, brightness, and applications, the histogram based histogram is adopted in this work.

There are many methods for selecting threshold using image histogram such as Ridler and Calvard algorithm, Tsai algorithm, Otsu algorithm, Kapur et al. algorithm, Rosin algorithm, and the Triangle algorithm. In this work, the triangle threshold algorithm is used because almost all the blank background image have a clear peak, so we can select the threshold using this algorithm easily and give accurate values. Another reason for using this algorithm, sometimes the background of the image is darker than the object itself, so by detecting the slope of the hypotenuse of the triangle we will know which is darker the objects or the background.

A line is constructed between the maximum value of the histogram (peak) and the lowest value. The distance d between the line and the histogram h[b] is computed for all values of b from b = bmin to b = bmax. The brightness value b0 where the distance between h and the line is maximal is the threshold value. This technique is particularly effective when the object pixels produce a weak peak in the histogram as shown in Figure 1. We consider that the most repeated colour over the blank background image is the background colour.

3.3 Object Boundaries Detection

The slope of hypotenuse in the triangle threshold algorithm is used to determine the pixels belonging to the background. If the slope is positive, the object is darker than the background and the background consists of pixel with values greater than the threshold and the other pixels will be considered as objects points, and vice versa.

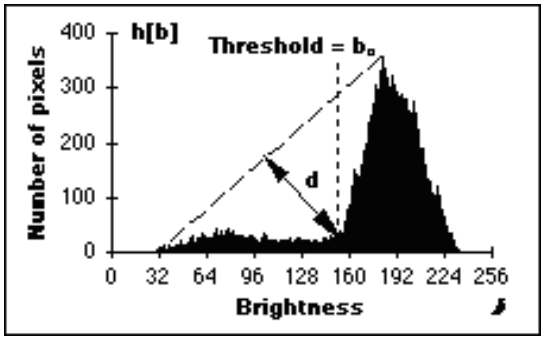

FIGURE 1: Triangle Thresholding Algorithm [10].

The object detection is done by scanning the image from top to bottom, left to right until we reach a point belonging to an object. Then, we stop scanning and start detecting the contour of the object by checking the eight adjacent pixels of the current pixel to move to the next point of the border.

After comparing the next pixel with the threshold, if this pixel does not belong to the object then go to the next direction in counter clockwise direction as shown in Figure 2; else add this pixel as a part of the object and move the current pixel to it. This operation continues until we reach the starting point.

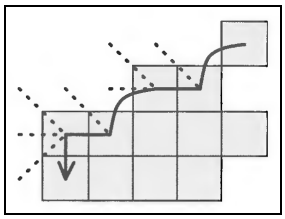

FIGURE 2: Boundary Tracing in 8-Connectivity [11].

3.4 Object Extraction

The previous step deals with the grayscale image and the threshold. It returns an array of points (closed area) representing the border of the object. Using the detected contours points, we crop the corresponding object from the original image. The detected object is extracted from the original image and then is stored as a transparent image which has only the detected object.

3.5 Image Object Compression

The new image with the extracted object is compressed using the JPEG lossless compression technique. This allows us to have good compression ratio without changing the quality of the object detected in the previous step. Since there is no transparency in the JPEG compression, we fill the transparent part in the image with the background colour which we had selected in the previous steps.

3.6 Multiple Objects Detection (Object Subtraction)

The image may contain more than one object, so we need to remove the found object from the original image and rescan the image for new objects. Removing the detected object is done by replacing its pixels values by the background colour selected from the previous steps without changing the dimension of the original image. Then, we repeat the previous steps starting from searching for an object in the image, object extraction, image object compression until no objects found in the original image.

3.7 Compressed File Structure

The compressed image generated from the previous steps is then stored in file that has two main parts: header and body. The header of the file contains general information about the image such as average background colour, width and height of original image, the position of each object given by its top left point, its array size and the number of objects extracted.

The body of the file includes the objects extracted. It contains the information about each object detected stored as an array of bytes that will be converted into image in the image decompressing phase.

3.8 Process Scenario

In this chapter we will make an example on our technique by choosing an image; this image is a human image as shown in figure 3 (a), first of all we detect the contour of the main object as shown in figure 3 (b), the main object is the human himself, then we extract the main object from the image so it will result two new images the background and the main object as in figure 3 (c), figure 3 (d) then we compress the image in figure 3 (d), we choose the average background color, after that we store all these information as a text as shown in figure 3 (e), this text is considered as the final output from the compressing process.

After we compressed the file we need to retrieve the image (decompress the stored file), in figure 3 (f) show the image after decompression process, it is same as original in height, width and the main object, the only deference is the background color is shown as one solid color.

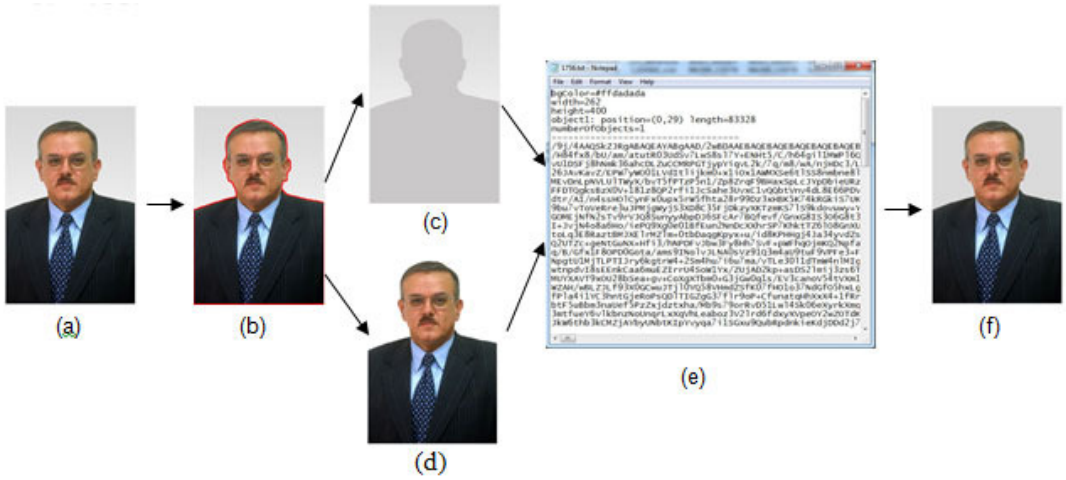

FIGURE 3: Example Scenario.

4. ALGORITHM EVALUATION

An evaluation of the abilities and the features of this technique are presented in this section. The evaluation aims at comparing the results of the proposed technique with other compression techniques in term of quality, compression ratio and time.

4.1 Dataset

To test this application we used a large set of images from the student's registration department in Princess Sumaya University for Technology. Because this kind of applications considers the background as irrelevant information.

The dataset consists of 1000 student images of type bitmap BMP. They contain mainly the face of the student (the main object) and a blank background with different colours, the maximum file size is 0.926 megabytes and the minimum size is 0.111 megabytes, with a total size 317 megabytes.

4.2 Benchmarks

We will evaluate our system in term of the following benchmarks:

- The quality of the significant information of the image (the main objects in the image)
- Compression ratio (the size of generated image after executing the compression comparing with the original image).
- The time elapsed to compress the images.
- The time elapsed to decompress the images.

The proposed compression technique is compared with the following compression algorithms: Lossless compression using JPEG algorithm, Lossless compression using LZW algorithm with TIF file format and Lossy compression using GIF algorithm. These techniques are chosen based on their popularity and their effectiveness in all types of images.

4.3. Results

4.3.1 Quality of Main Objects In The Image

We compared the resulted images after compression process with the original images using an application that checks each pixel in the original image with the same pixel in the compressed image; the results are shown in table 1. We noticed that the quality of the main objects in the image is preserved in the JPEG, TIF, and in our system. But when we used the GIF compression the quality is reduced since it is lossy compression.

JPEG	TIF	GIF	Proposed Technique
Yes	Yes	No	Yes

TABLE 1: Quality of Compressed Main Objects.

4.3.2 Compression ratio

The compression ratio (CR) measure the amount of compression realized. Different definitions are available in the literature. We considered the following definition for the purpose of evaluation: CR=(Compressed image size/ Original image size)*100%

Table 2 shows the size of dataset images before and after compressing the images using the different compression techniques. Table 3 highlight the compression ratio obtained from the different techniques applied on the 1000 image of the dataset.

Number of images	BMP	JPEG	TIF	GIF	Proposed Technique
10	3.133	2.117	2.564	0.548	0.884
100	31.105	21.094	25.452	5.361	8.568
500	159.694	108.761	131.497	27.308	44.957
1000	317.198	214.327	259.248	54.235	91.328

TABLE 2: Compressed Images Sizes in MB.

JPEG	TIF	GIF	Proposed Technique
67.5%	**81%**	**17%**	**29%**

TABLE 3: Compression Ratio Evaluation.

The GIF algorithm got the best compression ratio because it is a lossy compression, the quality of the compressed image is not the same as the original image and there is some information lost. The proposed technique got the best results over the lossless techniques JPEG and the TIF techniques, because our algorithm eliminate the background then compress the image of main objects with the lossless JPEG compression. Thus, the amount of data needed to represent the image is less than the JPEG and the TIF algorithms which store all the image information including the with background details.

4.3.3 Compression Time

The time elapsed to compress the 1000 images is different according to the algorithm used. Our algorithm needed more time than the other techniques. It consume more time in the preliminary steps such as: the grayscale image generation, the histogram generation, the average background selection, object detection. As this cost is paid only once, the other metrics mainly the compression ratio favorites the proposed technique.

5. CONCLUSION

A technique for compressing the image is proposed and evaluated in this paper. It saves the quality of the main components of the image and achieves good results in term of compression ratio. This technique is adequate for applications where the background is not important in the image.

Image compression is very important for image storage and transmission, so as future work we plan to experiment our system under different environments, illumination change, variable noise levels, and study the effects of image pre- and post-filtering. Special cases such as images with background colour very close to the objects colour need to be investigated.

6. REFERENCES

[1] R. C. Gonzalez and R. E. Woods, Digital Image Processing, (3rd edition). New Jersey: Prentice Hall. 2008.

[2] R. Al-Hashemi and I. Wahbi Kamal, A New Lossless Image Compression Technique Based on Bose, Chandhuri and Hocquengham (BCH) Codes. International Journal of Software Engineering and Its Applications, Vol. 5 No. 3. 2011.

[3] Sonal, D. Kumar, A Study of Various Image Compression Techniques. COIT, RIMT-IET. Hisar. 2007

[4] The National Archives, Image compression Digital Preservation Guidance Note 5. Surrey, United Kingdom. 2008.

[5] O. AL-Allaf, Improving the Performance of Backpropagation Neural Network Algorithm for Image Compression/Decompression System. Journal of Computer Science. Vol. 6 No. 11. 2010.

[6] K. Ramteke and S. Rawat, Lossless Image Compression LOCO-R Algorithm for 16 bit Image.2nd National Conference on Information and Communication Technology, Proceedings published in International Journal of Computer Applications. 2011.

[7] D. Chakraborty and S. Banerjee, Efficient Lossless Colour Image Compression Using Run Length Encoding and Special Character Replacement. International Journal on Computer Science and Engineering, Vol. 3 No. 7. 2011.

[8] Canadian Association of Radiologists, CAR Standards for Irreversible Compression in Digital Diagnostic Imaging within Radiology. Ontario, Canada. 2011.

[9] M. Sezgin and B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging, Vol.13, No. 1. 2004.

[10] H. Zhou, J. Wu and J. Zhang, Digital Image Processing: Part II, (2nd edition). London: Bookboon. 2010

[11] M. Sonka, V. Hlavac and B. Roger, Image Processing, Analysis, and Machine Vision, (3rd edition). India: Cengage Learning. 2007.