A Study for Image Compression Using Re-Pair Text-based Algorithm

Pasquale De Luca*[1][2], Vincenzo Maria Russiello[2], Raffaele Ciro Sannino[2] and Lorenzo Valente[2]

[1]Department of Science and Tecnhologies, University of Naples "Parthenope", Naples I-80143, Italy
[2]Department of Computer Science, University of Salerno, Fisciano I-84084, Italy
Email: p.deluca16@studenti.unisa.it

Abstract Compression is an important topic in computer science which is allowing us to store a larger amount of data on our data storage. There are several techniques to compress files. In this essay, we will describe the most important algorithm to compress images, which is JPEG, and we will compare it to another method, in order to provide solid arguments as to why JPEG should not be used for images. The most known encoding technique to compress texts is the Human Encoding, which will be explained in detail. We will illustrate the way in which we are able to use a method of compressing texts for compressing images and explaining in detail the method and reasoning behind choosing a particular format for the images rather than any other formats.The Re-Pair algorithm is the method studied and analysed in this essay.This algorithm was used solely for compressing grammatical contents. At the end of this essay we shall prove that using this method will give the best results.

Keywords: Image compression, Re-Pair, compression, BMP

1 Introduction

This essay is a deep study on whether it is possible to execute and use a particular compression algorithm on images. Aside from the canonical compression algorithms for images such as JPEG [1] and PNG, this work has aimed to show how to apply the Re-Pair algorithm, used purely for text compression, on compressing images. We looked for similar or related work in literature, but the research had an inconclusive outcome, since we were only able to find papers and work done on Re-Pair applied on text. We will show a way of executing this algorithm on images using methods and techniques in order to generate a good input: The Re-Pair algorithm, which is used only for text segmentation, is structured according to canonical grammar rules. A brief analysis of the problem on the compression of images shall be carried out, using several and recognized algorithms. Moreover, we will describe the method used to compute this technique in an efficient way for it to be used for future works. Great importance will be given to compression methods for images and the comparison between the different techniques. We will explain which method is the best and how using it allows us to have a very efficient result generated by operation of the Re-Pair algorithm.

2 Methods

In this section we will show how the method can quantify Re-Pair algorithm on images and the techniques used to convert an image from any graphical format in BitMap format. Furthermore, we will explain why we used this format. We can classify compression algorithms in two groups or rather in lossless and lossy [2].

2.1 Lossless Compression Methods

Lossless compression algorithms are computing compression methods which are allowing the original data to be perfectly reconstructed from the compressed data; it is possible to compress an image quickly, without losing graphical or data information. Lossless algorithms such as Huffman coding[5], which belongs to Entropy Encoding subfamily, is the most used compression method on which a lot of compression algorithms are based, in particular JPEG [4]. This method enables us to compress an image by opening it in binary mode and reading a single byte as ASCII symbol and then applying Huffman Encoding in order to generate a compressed version of the raw image. Other algorithms that belong to lossless family are:

- PNG;
- TIFF;
- TGA;
- BGP;

There are other methods, but we chose to mention only the most important ones.Indeed, we have found other techniques which are used on monochromatic images such as RLE [3] and Chain Codes [6].

2.2 Lossy Compression Methods

The aim of these methods is compressing an image by choosing which parts of the information are discarded, starting with reducing the colour space such as Chroma Subsampling method and any Transform coding which belongs to an important transform such as Discrete Cosine Transform [7].

The following figures represent various applications of a DCT on images:

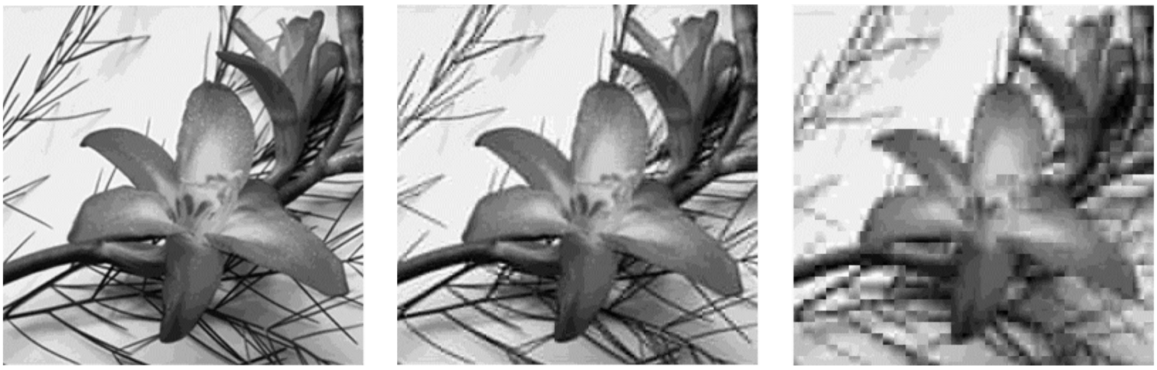

Figure 1. DCT compression step

In Figure 1 we see the differences among three compressed images. The first image is the raw, the second has 8 coefficients of compression and the last image has 64 coefficients of compression; The number of coefficients determine the percentage of compression and the loss of information. The more coefficients we have, the greater the loss of information becomes. Over time these two families were combined, but with minimal final results [8]. The methods previously mentioned have been developed for images in particular; now we will describe how to configure and use Re-Pair algorithm on images [8].

2.3 Re-Pair on Images

Re-Pair is an efficient grammar compressor that operates by recursively replacing high-frequency character pairs with new grammar symbols. The main approach consists in replacing the most frequently occurring pairs and adding a dictionary entry for each pair, mapping the new symbol to the replaced pair [10]. We will now briefly describe the steps of the algorithm: 1. The first step consists in computing the array sequence, which is an array structure where each entry consists of a symbol value and two pointers used to create doubly linked lists between sequences of identical pairs. 2. The second step consists in building an active pair table. This is a hash table from a pair of symbols to a pair record, which is a collection of information about a specific pair. The Re-Pair algorithm starts counting the pairs in input using a flag to indicate that a pair has been seen and if the pair is encountered again, a pair record is created.The

Re-Pair algorithm works mainly on text, on correct grammar-based texts in particular; for this reason, it is necessary to introduce the Context-Free Grammar, now CFG.

For definition:

A context-free grammar is a collection of rules of the form:

$$x_1, x_2, x_3, \ldots, x_k$$

where $x_1, x_2, x_3, \ldots, x_k$ are either terminal symbols (letters in the alphabet) or symbols that appear on the left-hand side of some rule.

To execute the Re-Pair algorithm on an image, we must convert the images to BMP Format because the common formats, such as PNG and JPEG and other already compressed formats, do not let us perform the compression. On the contrary, a BMP or rather Bitmap Image File is a simple format where there is not a high level of compression and its size on hard disks is greater than any image of different format. We tried to execute Re-Pair algorithm on JPEG and PNG images, but the results were not good, as shown in the following table:

Table 1. Execution of Re-Pair algorithm on different images format

Name file	Type image input	Size (Kb)	Size output (Kb)
hello	JPEG	67	72
test1	JPEG	192	388
test1	PNG	86	96
test2	PNG	155	227

The above table clearly demonstrates that is not possible using previously compressed images, as it creates confusion and redundancy.

2.4 Convert to BMP Format

The main step in applying this compression technique is converting any image file to BMP format since it is a simple format with low compression rate and it is allowing the compression algorithm to be efficient in such a way that the redundancy in bitmap picture assures a better compression [12].

2.5 Convert BMP to ASCII

In order to have an efficient compression with good results, we recommend converting the bitmap image to ASCII file using a simple method with any programming language, in order to open the image file in binary mode [13] and reading a chunk of octet bits after converting the octet of bits in decimal number. This step is optional. This table shows the conversion of bits to decimal to ASCII code [14].

Table 2. ASCII Table example

Letter	ASCII Code	Binary
a	97	01100001
b	98	01100010
c	99	01100011
d	100	01100100

In Table 2 we can see the Binary code of the first four numbers of ASCII table, represented by octet of bits and then transformed into decimal numbers, therefore is possible to associate the relative ASCII code using a cast of values. We can also see the results after the conversion.

3 Results

In this section we will briefly describe the results obtained after converting the images using the previous method. Our techniques consist in reading the BMP image using *zig-zag* reading techniques. This way there are more possibilities of obtaining an efficient result. The following table shows the results of Re-Pair algorithm executed on BMP images:

Table 3. Re-Pair algorithm on BMP images

Name	Size (Mb)	Size out (Mb)
hello	4,4	1,5
Ray	1,1	0,4
Lena	2,1	0,9
binary_ ex	1,0	0,3

In Table 3 it is shown that the compression rate is around 70 per cent, on monochromatic images in particular, because the redundancy of chroma pixel during the conversion to ASCII generates the same patterns, which are the best input for the Re-Pair algorithm.

3.1 Experimental tests

In order to highlight the ***compression rate*** obtained an execution of our methods has been done on four different *PNG* images.

Figure 2. First image

Figure 3. Second image

Figure 4. Third image

Figure 5. Fourth image

We propose a numerical tests based on *compression rate* as the final decompression returns a excellent rebuilding of image. The following table reports the substantial differences in terms of compressed file size:

Table 4. Compairson between PNG and Re-pair compress

Name file	Originale size [BMP]	PNG Compression	Re-Pair compression
first	731 kb	332 kb	136 kb
second	5883 kb	3982 kb	1921 kb
third	832 kb	431 kb	298 kb
fourth	1492 kb	824 kb	626 kb

In Tabel 4 we can note the high compression rate obtained by Re-Pair algorithm using the technique described in 2.4.

4 Discussion

Our study and final conclusions have shown that using a text compression method on images will achieve a higher compression rate than canonical compression methods such as *PNG*. Based on the results recorded in Table 3 we can give a short comment on the outcome of the compression of each image. The reason why the compression rate is particularly high compared to the other algorithms found in literature, which are used only for images, is due to the fact that this technique does not use frequency search which will keep the images as clear as possible. Our method described in 2.4 allows to obtain a higher increase of performance than the application of *canonical Re-Pair* executed in 2.3. The sole aim is obtaining a higher compression rate, even if the output's quality is reduced. The conversion of any image file to BMP and, afterwards, to ASCII, allows Re-Pair to obtain good results even if the detection of the pairs in the text, following the structure of the algorithm, can cause some problems. A new technique can be added to these steps, consisting in splitting the image in three different images: one for each colour channel or rather RGB. Subsequently, the Re-Pair algorithm must be applied on these files using a Discrete Cosine Transform reading, in order to have a better classification which will accommodate the redundancy in the starting file. An excellent idea for future endeavours can be crossing over this compression of images technique in losseless mode in order to achieve a greater efficiency in terms of visibility and compression.

References

1. Liu, Xianjin and Wei, Zhuo and Zhang, Qin and Huang, Jiwu and Shi, Y.Q.. (2019). Downscaling Factor Estimation on Pre-JPEG Compressed Images. IEEE Transactions on Circuits and Systems for Video Technology. PP. 1-1. 10.1109/TCSVT.2019.2893353.
2. Xu, Juncai, et al. "SSE Lossless Compression Method for the Text of the Insignificance of the Lines Order." arXiv preprint arXiv:1709.04035 (2017). APA
3. Robinson, A. H., and Colin Cherry. "Results of a prototype television bandwidth compression scheme." Proceedings of the IEEE 55.3 (1967): 356-364.
4. Chang, Y-W., T-K. Truong, and Y. Chang. "Direct mapping architecture for JPEG Huffman decoder." IEEE Proceedings-Communications 153.3 (2006): 333-340.
5. Huffman, David A. "A method for the construction of minimum-redundancy codes." Proceedings of the IRE 40.9 (1952): 1098-1101.
6. Raj, Y. Arockia, and P. Alli. "Pattern-based Chain Code for Bi-level Shape Image Compression." (2018). APA
7. Ahmed, Nasir, T_ Natarajan, and Kamisetty R. Rao. "Discrete cosine transform." IEEE transactions on Computers 100.1 (1974): 90-93.
8. Raid, A. M., et al. "Jpeg image compression using discrete cosine transform-A survey." arXiv preprint arXiv:1405.6147 (2014).
9. Wahba, Walaa Z., and Ashraf YA Maghari. "Lossless Image Compression Techniques Comparative Study." International Research Journal of Engineering and Technology (IRJET), e-ISSN (2016): 2395-0056. APA
10. Yamashita, Yoshiyuki, and Ikuo Nakata. "Coupled context-free grammar as a programming paradigm." International Workshop on Programming Language Implementation and Logic Programming. Springer, Berlin, Heidelberg, 1988.
11. Larsson, N. Jesper, and Alistair Moffat. "Off-line dictionary-based compression." Proceedings of the IEEE 88.11 (2000): 1722-1732.
12. Anand, Anjali. "BMP To JPEG-the conversion process." Journal of Global Research in Computer Science 2.6 (2011): 145-150.
13. Joshi, Keshav. (2018). A New Approach of Text Steganography Using ASCII Values. International Journal of Engineering and Technical Research. 7. 490-493.
14. Cahn, Robert S. "ASCII protocol conversion revisited." IEEE Journal on Selected Areas in Communications 8.1 (1990): 93-98.