# Image Compression Through Combination Advantages From Existing Techniques

**Elda Cina** *eldadoda@uamd.edu.al*
*Information Technology Faculty*
*"Aleksander Moisiu", University*
*Durres, Albania*

**Esmerald Aliaj** *esmeraldaliai@yahoo.com*
*Information Technology Faculty*
*"Aleksander Moisiu", University*
*Durres, Albania*

**Habib Hamam** *Habib.Hamam@umoncton.ca*
*Faculty of Engineering, University of Moncton, Canada*
*School of Engineering, Canadian Institute of Technology, Albania*

**Abstract**

The tremendous growth of digital data has led to a high necessity for compressing applications either to minimize memory usage or transmission speed. Despite of the fact that many techniques already exist, there is still space and need for new techniques in this area of study. With this paper we aim to introduce a new technique for data compression through pixel combinations, used for both lossless and lossy compression. This new technique is also able to be used as a standalone solution, or with some other data compression method as an add-on providing better results. It is here applied only on images but it can be easily modified to work on any other type of data. We are going to present a side-by-side comparison, in terms of compression rate, of our technique with other widely used image compression methods. We will show that the compression ratio achieved by this technique tanks among the best in the literature whilst the actual algorithm remains simple and easily extensible. Finally the case will be made for the ability of our method to intrinsically support and enhance methods used for cryptography, steganography and watermarking.

**Keywords:** Digital Data, Compression Techniques, Compression Rate, Data Security.

## 1. INTRODUCTION

Nowadays almost all information has become digital. The amount of information which is stored and processed every day has grown tremendously over the past few years and the rates with which it is continuing to grow are increasing rapidly. As a result, despite the improvement of hardware and transmission medium capacity, the performance of resource usage has been lowered. This is why data compression is as relevant today as it has ever been, due to the need to minimize bandwidth usage and speed for data transmission to and from ATMs, satellites, cellular phones, etc.

Data compression could be attained in two forms: lossless and lossy. Lossy compression can be achieved more easily and give better compression ratios due to the limitations of human vision. Lossless compression on the other hand is harder to achieve, but is very important on specific data types such as text, where each single letter has to be reproduced identically as in the source file, or in specific disciplines like astronomy or medical imagery, where the loss of a single pixel may alter results and lead to inadequate decisions. This high necessity for data compression makes it a highly demanded research field, regardless of the fact that many successful techniques already exist. Within this paper we are going to present a new technique for Image compression conceived by Prof Hamam [1] and developed by our team. This technique is used for the first time in the field of data compression and security, and yields significant advantages compared to most preexisting techniques. Its main scope is lossless compression but it can also be converted for use with lossy compression with minimal effort.

This paper is comprised of six sections starting with a general introduction, and followed by a summary of related works. The third section is devoted to our approach of image compression, and the fourth section will present the experimental results. The fifth section is a discussion about data representation through combinations versus existing techniques, and lastly, concluding remarks will be given in section six.

## 2. RELATED WORK

As we mentioned in the above section many good techniques already exist for data compression. With this paper we are going to discuss only one type of data which is image.

The lossy techniques are not of interest to this paper. Some of the most used techniques for lossless compression are [2-7]:

- ✓ Run length encoding (RLE): It is the simplest compression method used for sequential data. It replaces sequences of identical symbols (pixels), called runs by shorter symbols.
- ✓ Huffman encoding: It is a general technique for coding symbols based on their statistical occurrence The symbols (pixels) that occur more frequently are assigned a smaller number of bits, while the symbols that occur less are assigned larger number of bits.
- ✓ Lempel-Ziv-Welch coding (LZW): is a dictionary based coding which can be static or dynamic. In static dictionary coding, dictionary is fixed during the encoding and decoding processes. In dynamic dictionary coding, the dictionary is updated on fly.
- ✓ Arithmetic coding: is a form of entropy encoding. A sequence of characters is represented by a fixed number of bits per character, as in the ASCII code. Frequently used characters will be stored with fewer bits and not-so-frequently occurring characters will be stored with more bits. This leads to fewer bits used in total.
- ✓ Delta encoding: also called Delta compression and it is a way of storing data in the form of differences between sequential data rather than complete files. The most typical application is Delta encoding in HTTP, which proposes that HTTP servers should be able to send updated Web pages in the form of differences between versions (deltas). Pages change slowly over time, rather than being completely rewritten repeatedly. This leads to decreased Internet traffic.
- ✓ Discrete Cosine Transform: is a frequency domain technique which decorrelates neighboring pixels through cosine manipulations. It is the basis of JPEG standard.
- ✓ Discrete Wavelet Transform: is also a frequency domain technique which uses high and Low pass filtering to obtain compression. It is the basis of JPEG2000 standard.
- ✓ Fractal image compression: is a recent image compression method which exploits similarities in different parts of the image. The algorithm begins with the complete image, and then splits the image into a number of smaller blocks.

These basic techniques have been the object of studies for many researchers. Further development and intersection of these techniques has yielded many compressing applications like: WinZip, WinRar, JPEG and JPEG200 standards, PAQ and ZPAQ, BCIF, BMF, RawSpeedCmp (RawSpeedCompress2), [8-10] etc.

## 3. COMPRESSION THROUGH PIXEL COMBINATIONS

While existing techniques tend to decorrelate the relation between neighboring pixels, data representation through combinations uses this as an advantage to achieve better compression ratios. According to data representation through combinations, instead of defining the signal by an array of samples, it can be defined by the index (number) of the combination in the list of all possible combinations of sample values [1]. This enables flexible intrinsic compression and encryption opportunities.

In our case the list (or table) is built from all possible combinations of images of size MxN pixels, where each pixel has a value from 0 to L-1 (color level, for example gray level). The number of combinations is $L^{WxH}$, which means that the index of any image is between 0 and $\log_2 (L)^{WxH-1}$. If we consider L =256 (as the gray levels are), to store this index in a file, we need between 0 and 8 x (WxH) bits (where 8 x (WxH) bits is the worst case). The lower is the index, the lower is the number of bits needed.

Depending on the content of the image, the compression rate varies from:

$$\min = \frac{W \times H \times \mathrm{roundup}(\log_2(L))}{\mathrm{roundup}(W \times H \times \log_2(L))} : 1 \qquad \text{to} \qquad \max = W \times H \times \mathrm{roundup}(\log_2(L)) : 1$$

Where roundup(x) returns the next integer bigger or equal x.

While the images tend to be more qualitative they become bigger and bigger. Therefore the representative index, which grows exponentially with the size of the image, becomes too large to be processed. As a consequence, the image should be divided into smaller blocks. Below is described the main class of our technique implementation.

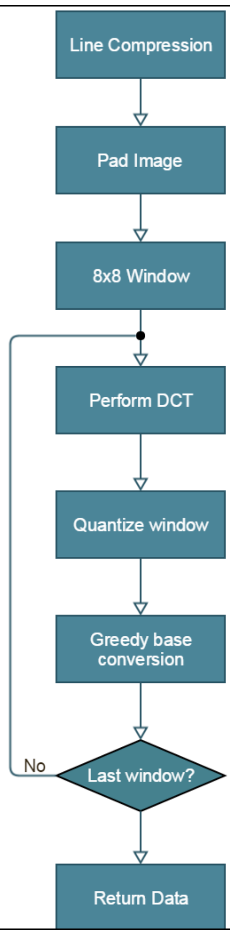



**FIGURE 1:** Dividing the image into groups.

**Algorithm 1** Line Compression

```
procedure GDC(Image)
    Pad Image
    for each window ∈ Image do
        win_sh = win-128
        Perform DCT on window
        Quantize resulting windows
        Vectorize windows
        Apply greedy grouping
        Convert Base to highest available
        Apply EOB
    end for
end procedure
```

**FIGURE 2:** Extracting the index algorithm and converting to indexes.

The first figure is the diagram of the function which divides the block into small groups and then the groups into the corresponding indexes. The second one is its pseudo code.

Referring to the proposed technique each image to be stored or shared over the web, needs to be represented by a file which contains a group of indexes. The indexes can have the fixed or variable length.

## 4. EXPERIMENTAL RESULTS

A number of experiments have been carried out in order to evaluate the performance of the proposed algorithm. It is applied into most known image benchmarks libraries like: Waterlo [11], Kodak Set, Image compression Benchmark [12], Maximum compression[13], Squeeze-chart 2015 [14]. For the purpose of simplicity, we are illustrating here only the results form SqueezeChart 2017 which is the most updated database (updated lastly on March 2016). It is a database which includes over 125 applications which test different data types, which include all the techniques mentioned at section II. As we are focused on images we can say that it contains 13 images of type .pgm and 25 images camera raw which were first converted to JPEG and then compressed by our technique.

| Image | Cr.Fix | Best Cr from Library | Times better | Best actual compressor from library |
|---|---|---|---|---|
| **Canon mac 3** | 4 | 1.2 | 3.333 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **EOS 6D** | 4 | 1.17 | 3.419 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **canon_eos_m_04** | 4 | 1.15 | 3.478 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **fujifilm_finepix_x100_11** | 4 | 1.83 | 2.186 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **fujifilm_x_e1_20** | 4 | 2.14 | 1.869 | RawSpeedCmp3 + Precomp –cn '+ MCM 0.4 -9 |
| **fujifilm_xf1_08** | 4 | 1.76 | 2.273 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **leica_m82_05** | 4 | 2 | 2.000 | RawSpeedCmp3 + ZPAQ method 6 |
| **leica_x1_10** | 4 | 1.9 | 2.105 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **nikon_1_v2_17** | 4 | 1.17 | 3.419 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **nikon_d4_10** | 4 | 1.3 | 3.077 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **nikon_d5200_14** | 4 | 1.18 | 3.390 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **olympus_epm2_16** | 4 | 1.1 | 3.636 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **olympus_om_d_e_m5_24** | 4 | 1.11 | 3.604 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **olympus_xz2_10** | 4 | 1.51 | 2.649 | DGCA solid |
| **pentax_k5_ii_12** | 4 | 1.39 | 2.878 | RawSpeedCmp3 + Precomp –cn + |

| | | | | |
|---|---|---|---|---|
| | | | | ZPAQ method 6 |
| **pentax_q10_19** | 4 | 1.68 | 2.381 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **sigma_dp2** | 4 | 1.22 | 3.279 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **sigma_sd1_merrill_13** | 4 | 1.27 | 3.150 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **sony_a55** | 4 | 1.25 | 3.200 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **sony_a77_08** | 4 | 1.36 | 2.941 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **sony_a99_04** | 4 | 1.39 | 2.878 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **samsung_nx20_01.srw** | 4 | 1.67 | 2.395 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **samsung_nx1000_19.srw** | 4 | 1.85 | 2.162 | RawSpeedCmp3 + Precomp –cn + ZPAQ method 6 |
| **Canon** | 4 | 2.86 | 1.399 | PAQ8pxd_v4 (19.04.2012) -7 -worldwide PAQ crew- |
| **Fuji** | 4 | 2.61 | 1.533 | ZPAQ 6.36 -method bmp_j4 -tiny -add arc in.bmp (2013) M. Mahoney (input=bmp) |
| **Hubble** | 4 | 1.71 | 2.339 | BMF v2.01 (24.05.2010) -S Dmitry Shkarin |
| **Oly 24** | 4 | 3.21 | 1.246 | .DNG/lossless-JPEG |
| **Sigma 24** | 4 | 4.03 | 0.993 | BMF v2.01 (24.05.2010) -S Dmitry Shkarin |
| **Radiograph** | 4 | 6.59 | 0.607 | PAQ8pxd_v4 (19.04.2012) -7 -worldwide PAQ crew- |
| **Sony 2** | 4 | 2.82 | 1.418 | RAW Image Format (from which the .TIFF here comes) |
| **Medical 1** | 4 | 3.698761 | 1.081 | .MRP Minimum Rate Predictors v0.5 -o -I 12 Ichiro Matsuda |
| **Medical 2** | 4 | 5.19288 | 0.770 | .MRP Minimum Rate Predictors v0.5 -o -I 12 Ichiro Matsuda |
| **Medical 3** | 4 | 2.785996 | 1.436 | .MRP Minimum Rate Predictors v0.5 -o -I 12 Ichiro Matsuda |
| **Medical 4** | 4 | 2.083985 | 1.919 | .MRP Minimum Rate Predictors v0.5 -o -I 12 Ichiro Matsuda |

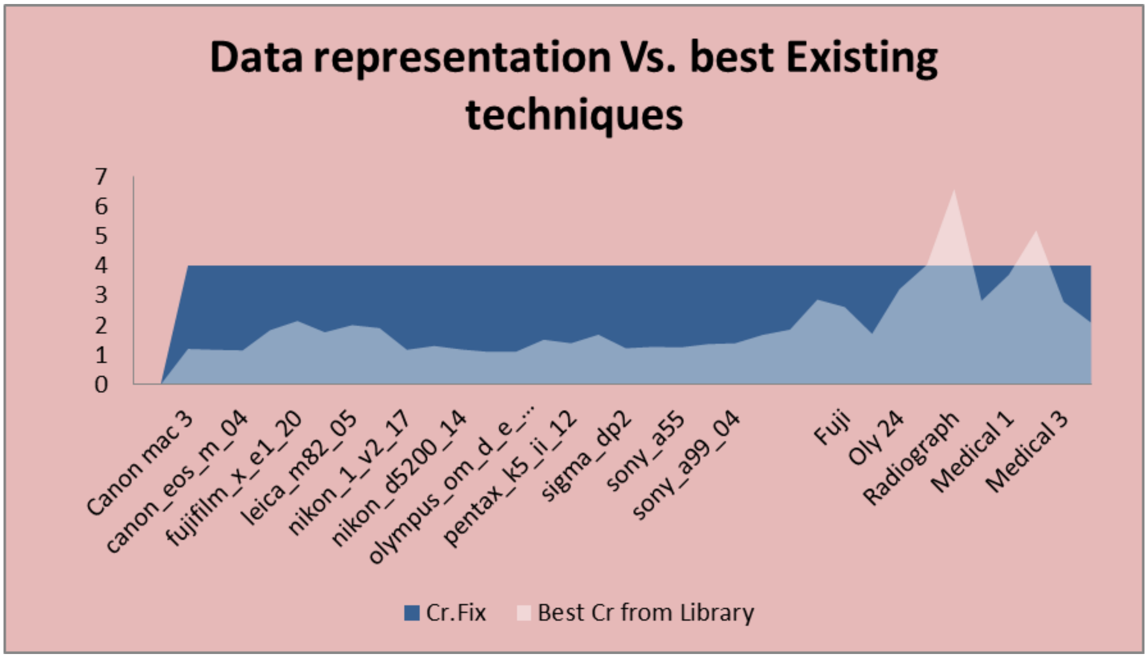
The comparison is made with the best result of the library for each given image. For example; CANON is best compressed by **PAQ8pxd_v4 (19.04.2012) -7 -worldwide PAQ crew- ,** while FUJI is best compressed with **ZPAQ 6.36 -method bmp_j4 -tiny -add arc in.bmp (2013) M. Mahoney (input=bmp).**

The table also shows how much better, is our technique compared with best actual technique.

As we mentioned in the section above, two different implementation are used for indexes, fixed and variable length. The fixed length indexes implementation gives a fix compression rate of 4 and is a faster implementation in general. The variable length indexes implementation uses an EOB which in some cases increases the length of the index. From the experiments done we can say that the EOB implementation is better to be used in those images who have regions with same color level, or the variability between pixels in terms of colors is very low. For example Radiography or Medial images form the table above.

## 5. DISCUSSION

As can be seen from the experimental results above, we obtain a compression ratio of around 4.




We considered two methods: fixed length of compressed block or delimitation of compressed blocks by an end-of-block (EOB) sequence. Currently the fixed length compression technique shows better results, thus it has been used for the comparisons table above. Compared with existing commonly used formats such as **RawSpeedCmp3** + **Precomp –cn** + **ZPAQ method 6** (or others listed in the libraries) are statistically better. It is said statistically because in a few cases like in imagery images (**Radiograph** and **Medical 2** where black level is commonly founded) our compression rate is slightly lower than the library. However, in these types of images, our approach with EOB sequence gives a factor of compression rate over 7, which is better than those from library. As such we can say that the study is not exhausted, and compression ratio may be further improved.

Again if we take into consideration the above results we can see that the average compression rate of all the techniques (which are supposed to be the best cases for each image) is round 2.1. Result which also leads to the fact that data representation through combination compression gives intrinsic advantages in terms of compression.

The comparison is made only in terms of compression rate because the new techniques introduced in this article are developed in MATLAB platform and as such are not in the optimum level of speed of execution. Therefore this issue needs to reconsider in the future developments.

Particular attention has been given to lossless image compression. Generally, lossless compression is required for texts since any modification of a text, no matter how small, is not acceptable because it may change the meaning of the text or make it non-understandable. As the image is of concern, a small alteration of some of the pixels is generally acceptable so long as it is not perceivable, or barely perceivable, with the naked eye. That said, in some cases, any modification of the pixels, no matter how small, is not allowed, for example if the image contains secrete data embedded by steganography or watermarking techniques.

## 6. CONCLUSIONS

We suggested a new method for image compression. We focused on lossless image compression since in the future we will combine our method to steganography and watermarking techniques. We obtained results that comparable with commonly used techniques such as RawSpeedCmp3 + Precomp –cn + ZPAQ method 6, JPEG versions, BCIF, etc.

As a new method, its power and perspectives are not exhausted yet by intensive research. However, this method opens several avenues of research. In the near future, we try to avoid fixing the block length and adding the supplementary bit sequence, called EOB, by considering the behavior of the curve of compressed values.

## 7. REFERENCES


[1] H. Hamam, “A new representation of data through combinations”, *ICCAT 2013*,

[2] Grgic, S. Mrak, M., & Grgic, M. “Comparison of JPEG image coders”. (2001) Retrieved October 23, 2003, from http://www.vcl.fer.hr/dtv/sgrgic_vip2001.pdf

[3] Dr. Charles F. Hall, “ A Hybrid Image Compression Technique,” Acoustics Speech & Signal Processing, , IEEE International Conference on ICASSP’ 85, Vol.10, pp 149- 152, Apr, 1985

[4] S.Navqi, R.Navqi,R.Riaz, F.Sidiqui "Optimized RTL design and implementation of LZW algorithm for high bandwidth applications". Electrical Review 2011 (4): 279–285.

[5] J. Rissanen, “Generalized Kraft Inequality and Arithmetic Coding”. IBM Journal of Research and Development 20: 198–203. doi:10.1147/rd.203.0198 (1976).

[6] S. Smith, “The Scientist and Engineer's Guide to Digital Signal Processing”, Elsevier Science, ISBN-13: 9780750674447 (2002)

[7] A. Papoulis, “The Fourier Integral and Its Applications”, McGraw-Hill, New York, 1962

[8] J.Miano, "Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP" Adison-Weley | 1999 | ISBN: 0201604434

[9] http://mattmahoney.net/dc/dce.html

[10] http://www.researchandtechnology.net/bcif/about.php

[11] https://ece.uwaterloo.ca/~z70wang/research/ssim/

[12] http://www.imagecompression.info/test_images/

[13] http://www.maximumcompression.com/

[14] http://www.squeezechart.com/