https://doi.org/10.2352/ISSN.2470-1173.2019.11.IPAS-263

Additional lossless compression of JPEG images based on BPG

Mykola Ponomarenko[a], Oleksandr Miroshnichenko[b], Vladimir Lukin[b], Karen Egiazarian[a]
[a] Tampere University, FIN 33101, Tampere, Finland;
[b] National Aerospace University, 61070, Kharkov, Ukraine

Abstract

The task of additional lossless compression of JPEG images is considered. We propose to decode JPEG image and recompress it using lossy BPG (Better Portable Graphics) codec based on a subset of the HEVC open video compression standard. Then the decompressed and smoothed BPG image is used for calculation and quantization of DCT coefficients in 8x8 image blocks using quantization tables of the source JPEG image. A difference between obtained quantized DCT coefficients and quantized DCT coefficients of the source JPEG image (prediction error) is calculated. The difference is lossless compressed by a proposed context modeling and arithmetical coding. In this way the source JPEG image is replaced by two files: compressed BPG image and the compressed difference which needed for lossless restoration of the source JPEG image. It is shown that the proposed approach provides compression ratios comparable with state of the art PAQ8, WinZip and STUFFIT file archivers. At the same time BPG images may be used for fast preview of compressed JPEG images.

Keywords: JPEG, JPEG additional compression, discrete cosine transform, context modelling

Introduction

JPEG standard [1] was proposed more than 25 years ago, but is still the most widespread and used method of lossy image compression. It is the main standard used for lossy image compression in digital cameras and mobile phones. A significant part of the modern Internet traffic is transmission of JPEG images during browsing web pages and using of cloud based image storages. Therefore, the task of additional compression of JPEG images is actual.

Many more efficient methods than JPEG are proposed recently, for example wavelet based standard JPEG2000 [2], discrete cosine transform (DCT) and partition schemes based ADCT [3]. The most efficient method of lossy image compression now is BPG [4] codec based on a subset of the HEVC open video compression standard [5]. These best modern lossy compression methods provide compression ratio (CR) by 1.5 … 1.7 times better than JPEG for the same quality of decompressed images [6]. These CR values are an upper bound of potential compressibility of JPEG images.

Due to large amount of JPEG images, which are transmitted in telecommunication channels, uploaded from smartphones to cloud storages and downloaded back, there is a necessity in an additional lossless compression of JPEG images. A possibility of an JPEG image decompression and more efficient compression of the image by a more efficient lossy image compression method is not acceptable in this case. The images before and after transmission should be the same, there should be no losses even they are visually undistinguished. Only lossless compression should be applied for additional compression of JPEG images.

Most efficient lossless compression of JPEG images are provided by PAQ8 archiver [7] и Stuffit archiver [8]. PAQ8 provides best additional compression of JPEG images (in average on 25%), but is very slow. STUFFIT a little bit (in average on 1%...2%) falls behind on PAQ8, but is much more faster.

Well known WINZIP archiver [9] in average provides additional compression on 18%...20%, and is slightly faster than STUFFIT. A method [10] based on AGU lossy compression [11] provides compression of JPEG images on 18%...20%, but is slower, than WINZIP.

A method of additional lossless compression JPEG based on RGC [12] is fastest from abovementioned methods, but provides additional compression only on 5% [13].

Most of such methods are based on more effective recompression of quantized DCT coefficients (see Fig. 1).

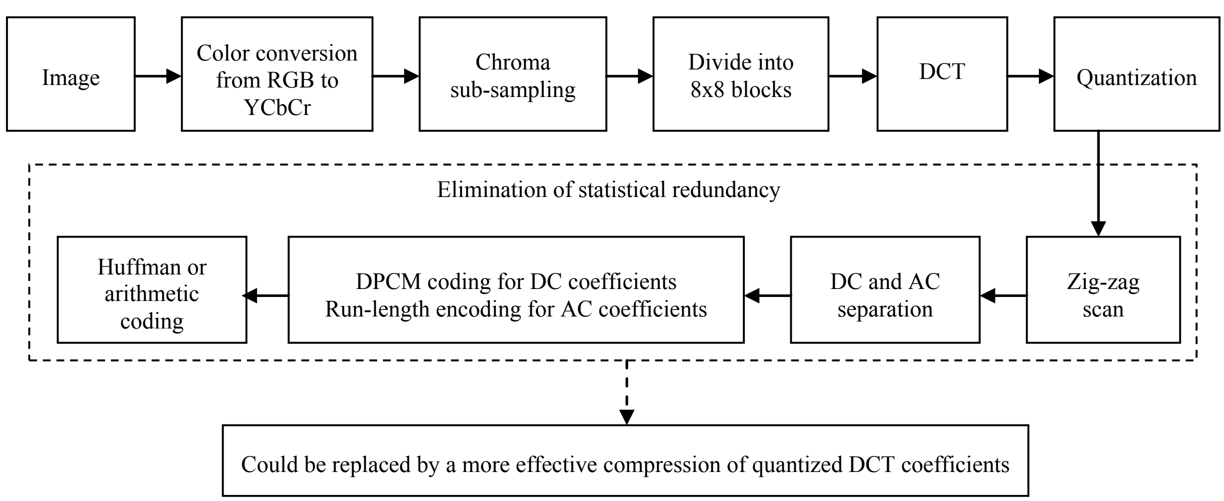


Fig. 1. Most used scheme of JPEG additional lossless compression

PAQ8 and STUFFIT compress JPEG images in average by 1.2...1.25 times. However, a practical usage of these methods (for example in cloud storages) is restricted by the necessity to decompress JPEG images before to view the images. It requires a large computing power, especially in the case of very slow PAQ8 archiver.

In this paper we propose a new method of additional lossless compression of JPEG images which produces two files: a BPG compressed image and compressed difference between the source JPEG image and the BPG image. In other words, the BPG image here is used for prediction of quantized DCT coefficients of the source JPEG image, and compression error of this prediction is compressed.

The output BPG image is also may be used for high quality preview (for example in cloud-based storages) of additionally compressed JPEG images without their decompression. The compressed difference between BPG and JPEG images guarantee lossless decompression of the source JPEG images (for example before downloading of the image from a cloud-based storage).

Description of the proposed method

If to decode an JPEG image and recompress it by BPG method with a similar quantization step (QS) then the resulting file will take significantly less memory. And the difference between decoded JPEG and decoded BPG images will be small.

Idea of the proposed method of additional lossless JPEG compression by prediction using BPG image (JACP) is to replace a source JPEG image onto two files: compressed BPG image and compressed difference between JPEG and BPG images (see Fig. 2). Here BPG image is used for prediction and better recompression of quantized DCT coefficients of source JPEG image as in Fig.1.

Let us explain the proposed method in details (see Fig. 3). JPEG image is partially decoded to quantized DCT coefficients of 8x8 pixel blocks. Let us denote the matrix of the quantized DCT coefficients as **A**.

Then the image is fully decoded and recompressed by BPG method. Let us denote this BPG image as **C**. The image is needed for prediction of values of A and may be used for a fast previewing of the source JPEG image.

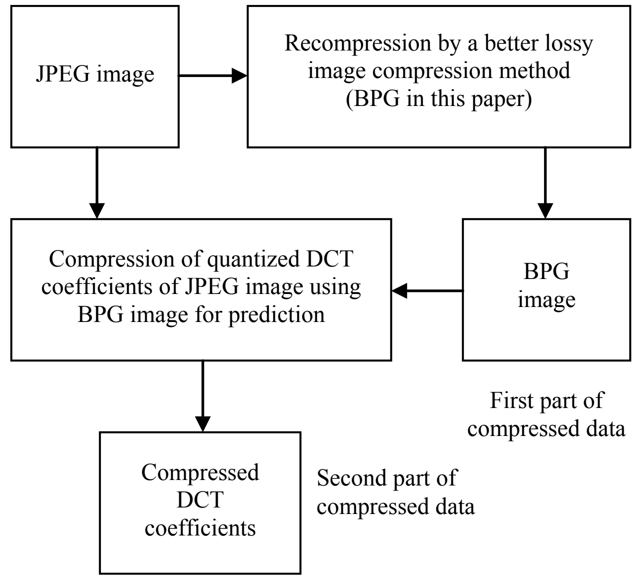


Fig. 2. Main idea of the proposed method

Than the image C is decoded and filtered by the method described in [15] and used in [3] with the threshold value equal to QS/2, where QS is quantization step used in BPG compression. The filtering decreases blocking artifacts which leads in decreasing of prediction error. This step can be eliminated from the proposed compression scheme with decreasing of compression ratio (CR) approximately on 0.5%.

Decoded BPG image is divided on 8x8 pixel blocks. For each block DCT coefficients are calculated. Let us denote the obtained matrix of DCT coefficients as B.

Due to losses introduced by BPG compression, matrix A and B/QS will differ.

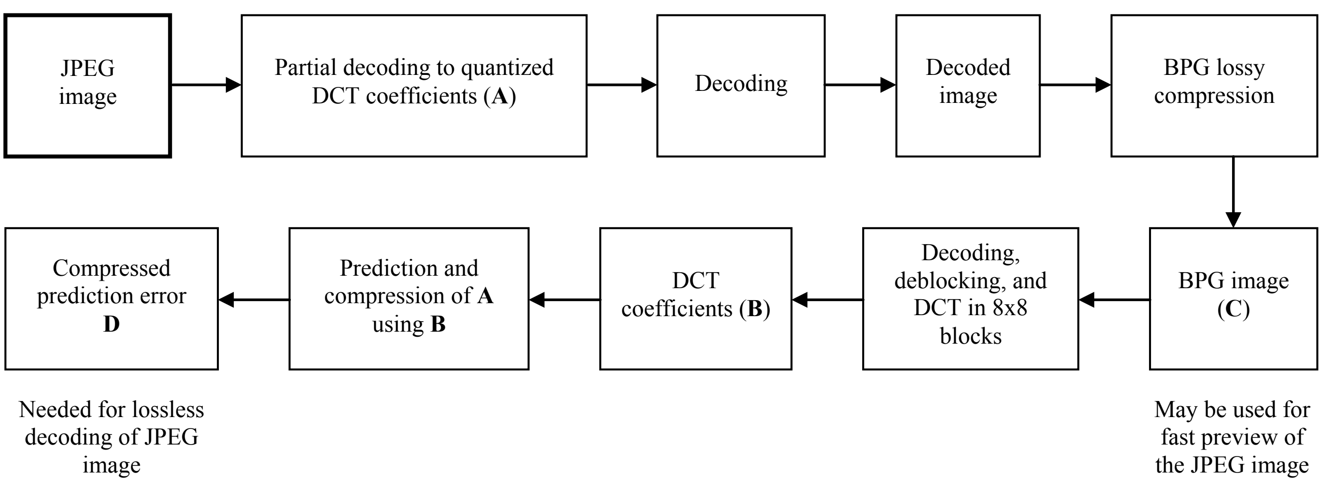


Fig. 3. Flow chart of the proposed additional lossless compression of JPEG images

To compress matrix **A** one can predict its values using values of matrix **B** and after this compress the prediction error which has significantly smaller entropy. For compression of the prediction error in the paper we use entropy coding on base of arithmetic coding and a sophisticated context modeling (it will be described in details in the next sections). One more subtask required a solution is selection of QS for the BPG compression. It is definitely should be proportional to quantization tables used in the compressed JPEG image.

An example illustrating the proposed method is shown on the Fig. 4.

a b c

*Fig. 4. An example of lossless compression of JPEG image by the proposed method: a) JPEG image, 33934 bytes; b) BPG image, 13503 bytes, c) difference between **A** and **B**/QS (grey pixels correspond to zero difference, bright pixels correspond to positive difference, dark pixels correspond to negative difference), 8836 bytes after context modeling and entropy coding.*

File of the source JPEG image takes 33934 bytes in memory or on a hard drive. Files of BPG image and compressed prediction error [A-B/QS] take in total 22339 bytes of memory and allow to decode of the source JPEG image without losses. For this image the proposed method provides additional lossless compression of the JPEG image in 1.52 times.

Flow chart of decompression of additionally compressed JPEG image is shown on the Fig. 5.

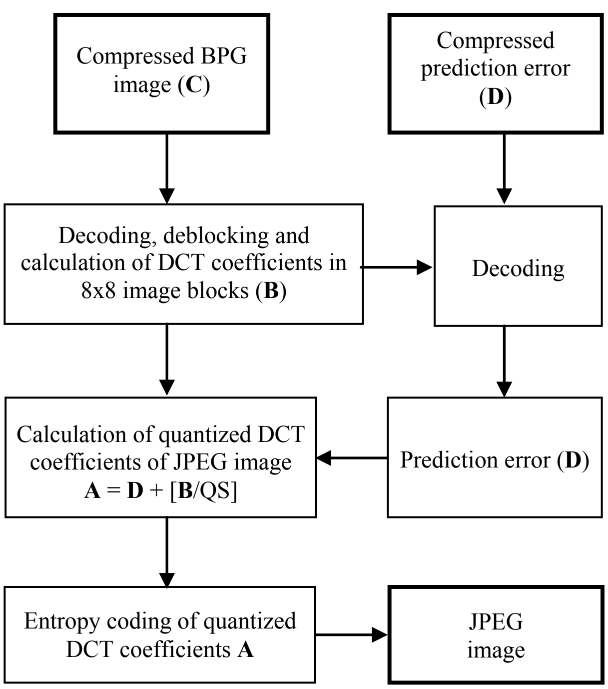


Fig 5. Flow chart of decoding of compressed JPEG image

As it seen from the scheme, compressed **C** and **D** allow to decode the matrix A without losses and than reconstruct the source JPEG images repeating JPEG entropy coding steps for quantized DCT coefficients.

Selection of quantization step for BPG

A bigger QS value for BPG compression will provide bigger CR for BPG image **C** and bigger prediction error **D** resulting in lower CR for the matrix **D**. On the contrary a smaller QS will provide lower CR for BPG image **C** and lower prediction error **D** resulting in bigger CR for the matrix **D**. Thereby selection of QS for BPG is an optimization task.

Let us take into accounting the following considerations for selection of QS for BPG compression:

1) Choosing of too small QS may result in C bigger than the source JPEG image which is inappropriate.

2) Selection of too big QS will lead to decreasing of size of C, but at the same time to increasing of absolute values of D.

3) Usage for BPG compression of the same QS as for the source JPEG image is not optimal too, because some of lowest bits of quantized DCT coefficients will be coded twice (in BPG image and in matrix **D**).

Given the above in the paper we will use quantization step for BPG method twice larger the one for source JPEG image. In this case most of higher bits of **A** will be precisely coded in the BPG image **C**. At the same time most of lowest bits will be coded only in compression of the difference **D**.

In out numerical analysis we will use JPEG compression with the same a small QS for each spatial frequencies and color components. It will simplify analysis and allow to use a BPG codec which has limited functionality (selection of only strength of compression instead of settings of arbitrary quantization tables).

Context modeling for prediction error compression

Let consider the proposed context modeling step by step on example of the image on Fig. 4.

The prediction error **D** is calculated as **D** = **A** - [**B**/QS], where [.] is operator of rounding to nearest integer value (see Fig. 4, c). If **D** is compressed directly by arithmetical coning then the size of compressed **D** file is equal to 14233 bytes. The result is far from optimal, because there is additional information (**B** matrix) which should be taken into accounting. Analyzing **B** values it is possible to divide compressed data into several classes. Aggregated entropy of these classes will be lower than entropy of whole matrix **D**. Thus compression of theses classes separately will result in decreasing of their total size in comparison to size of compressed matrix **D**.

Taking into accounting of absolute values of quantization

Let us calculate absolute values $\mathbf{Z_a}$ of **B** quantization as $\mathbf{Za} = |\mathbf{Z}|$, $\mathbf{Z} = \mathbf{B}/QS - [\mathbf{B}/QS]$. It is clear that $\mathbf{Z_a}$ values are in the range 0...0.5 (value of rounding).

Let us make the following hypothesis. If for a given DCT coefficient corresponding value of $\mathbf{Z_a}$ is close to zero (quantization error is very small), then most probably corresponding value of prediction error **D** is equal to zero. If for a given DCT coefficient the value of $\mathbf{Z_a}$ is close to 0.5 (quantization error is very big), then probability of $\mathbf{D} \neq 0$ for this DCT coefficient is relatively high.

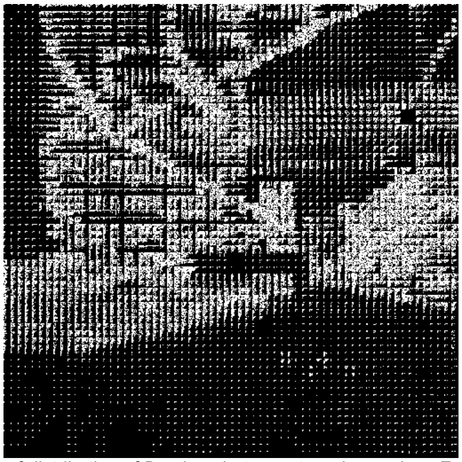


Fig 6. Map of distribution of D values between two classes by a Z_a threshold

Let us check this hypothesis dividing all **D** values onto two classes. Values of **D** with corresponded $\mathbf{Z_a}$<0.1 will be send to the class #1. Rest of values of **D** will be send to the class #2. Compression of the two classes (for the image on the Fig. 4) gives total size of both compressed files equal to 11355 bytes. It is significantly less than 14233 for direct compression of **D** values.

The map of distribution of **D** values between the two classes is shown on the Fig. 6. Comparing the map with the image Fig. 4, c one may note that the class #1 (dark pixels on the Fig. 6) is well corresponding to zero values of **D** (grey pixels on the Fig. 4, c), and the probability of **D**=0 for this class if much more higher than for the class #2 (bright pixels on the Fig. 6).

Let us note that it is possible to increase of compression effectiveness (compression ratio) by reasonable increasing of number of classes introducing of several threshold instead of one. In this paper 17 thresholds have been chosen as a result of numerical optimization. The 17 threshold allow to divide values of **D** matrix onto 18 classes (see the Table 1).

Table 1. Threshold for Z_a for splitting of D values on 18 classes

0.001	0.005	0.009	0.017	0.025	0.05	0.07	0.1	0.13
0.16	0.19	0.22	0.25	0.29	0.35	0.4	0.45	

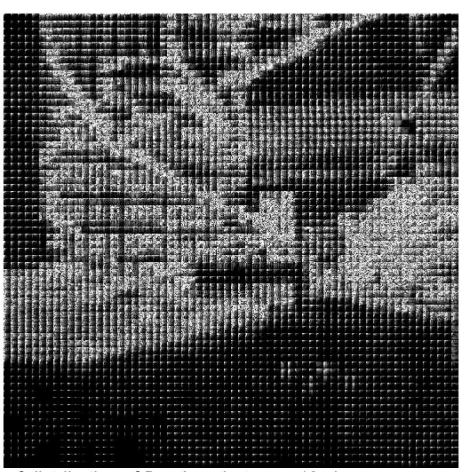


Fig. 7. Map of distribution of D values between 18 classes

The corresponding map of distribution of **D** values for JPEG image on the Fig. 4 between 18 classes are given on the Fig. 7.

This context modeling (dividing compressed data onto 18 classes) allows to increase compressibility of the data for this case from 11355 bytes to 10426 bytes.

Taking into accounting of sign of quantization

The context modeling proposed in the previous subsection allows to effectively predict probability of zero values of **D** matrix, but does not allow to predict signs of nonzero values. For prediction of the signs one should use signs of corresponding values of **Z**.

In the paper we propose to divide each from 18 classes formed in the previous subsection on two classes (corresponding to **Z**<0 and to **Z**≥0).

For better understanding the histogram of **D** values for the class #13 (0.22 ≤ $\mathbf{Z_a}$ < 0.25) of image Fig. 4 is given on the Fig. 8, a. Corresponding histograms after dividing this class on two new classes by sign of **Z** value are given on the Fig. 8, b and 8, c.

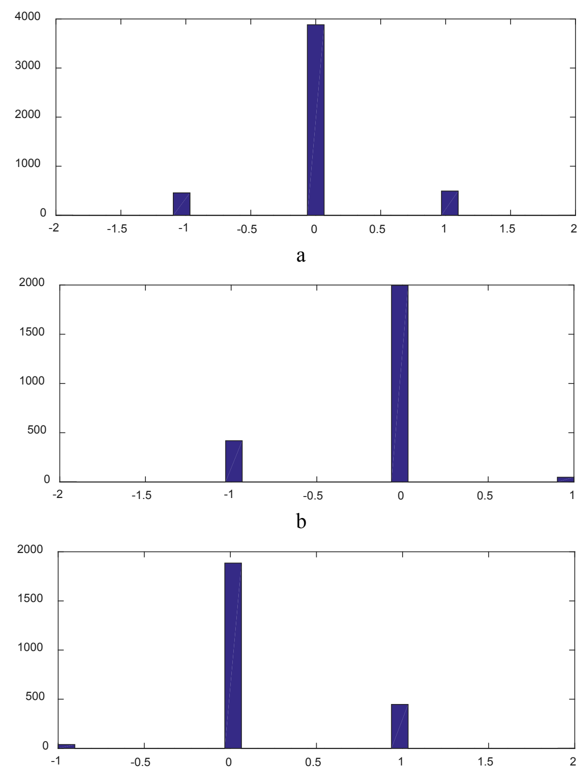


Fig 8. Histograms of D values for class #13 and for two new classes obtained by dividing of this class by sign of Z value.

It is well seen that there is approximately equal number of values -1 and 1 in the primary class (Fig. 8, a). In the same time in each of derived classes only one value is prevailing. Such classes have lower entropy and are more compressible.

This rule of dividing of a class on two classes is used for each form 18 classes. As a result 36 classes are formed which results in increasing compressibility of the data for Fig. 4 from 10426 bytes to 9044 bytes.

Taking into accounting of spatial information

Two kinds of context modeling proposed in previous sections do not take into accounting spatial information. However it is clearly seen on the Fig. 4, c, that pixels with high probability of zero values occupy compact regions (probability of zero value is higher if there are many zeroes in the neighborhood).

Let us take into accounting the spatial information in the following way. Map of spatial density **S** of $\mathbf{Z_a}$ values is calculated as:

$$S(i,j) = \sum_{k=i-1}^{i+1} \sum_{m=j-1}^{j+1} Z_a(k,l)\,, \qquad (1)$$

where i,j is coordinates of a DCT coefficient in $\mathbf{Z_a}$.

As a result of numerical optimization 18 thresholds for dividing of each class (before usage of signs of **Z** values) are given in the Table 2.

Table 2. Threshold for Z_a for splitting of D values on 18 classes

0.8	0.95	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Each of 18 classes is divided on two classes (for **S** value smaller and bigger than corresponding threshold). Than each class is divided on two classes by **sign** of Z value.

Thus as a result of all three kinds of proposed context modeling values of **D** matrix are divided on 72 classes, whose values are compressed by arithmetical coding separately. It allows to decrease size of compressed **D** file for the image on Fig. 4 from 14233 bytes to 8836 bytes.

Let us note that for further improvement of the context modeling one can uses already coded **D** values.

Numerical analysis using TAMPERE17 database

For numerical analysis 300 images (512x512 pixels) of database TAMPERE17 [14] are used. Comparisons with PAQ8, STUFFIT and WinZip are provided. TAMPERE17 is database of noise free images (σ of noise is less than 1) without interpolation, filtering and other processing (the database is available on www.cs.tut.fi/sgn/imaging/tampere17). All the images was preliminary compressed by JPEG standard with QS=8.

Histogram of CRs of additional compression of JPEG images of TAMPERE17 database is shown on the Fig. 9.

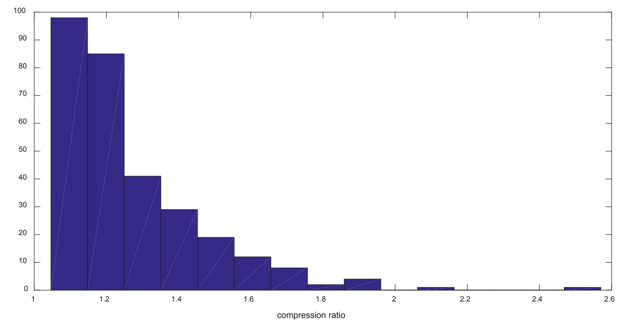


Fig. 9. Histogram of CR values for additional compression of JPEG images of TAMPERE17 database

It is clearly seen that most of CR values are on the level 1.2, while for some images CR exceeds 2.

Fig. 10 contains CR of additional lossless compression for compared methods averaged for all 300 images of TAMPERE17.

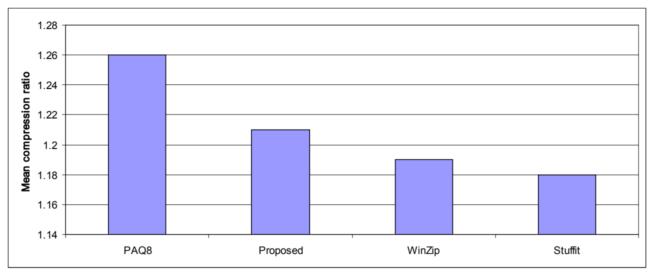
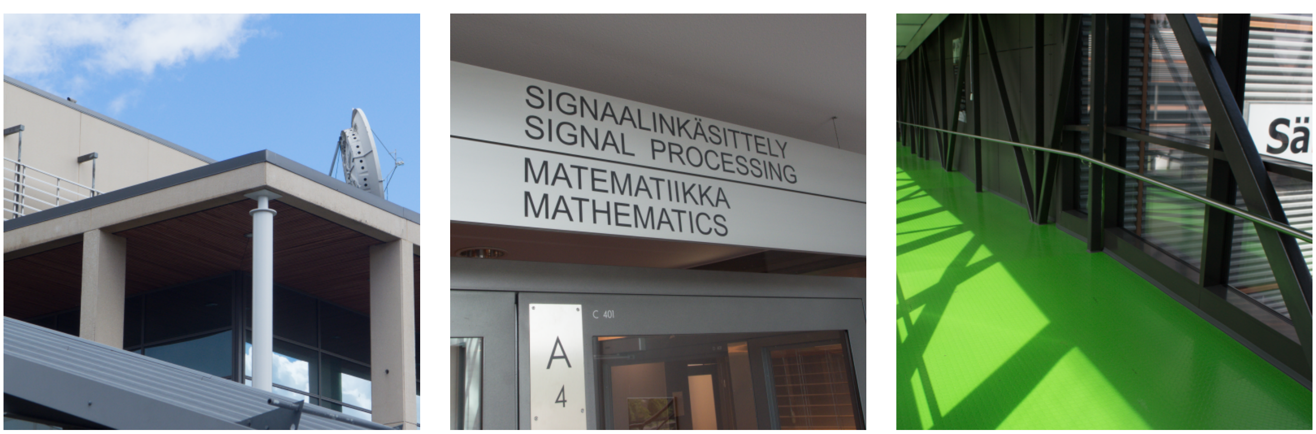


Fig. 10. Average CR for compression of JPEG images of TAMPERE17 by compared methods

The proposed JACP method provides average CR=1.21, giving behind of PAQ8 and outperforming fast WINZIP and STUFFIT archivers.

Let us note that for 25% of images of TAMPERE17 the proposed JACP method provides largest CR for all compared methods including PAQ8. Examples of such images are shown on the Fig. 11. All these images have a large homogeneous areas.



CR for PAQ8: 1.53, CR for JACP: 1.69 | CR for PAQ8: 1.55, CR for JACP: 1.75 | CR for PAQ8: 1.47, CR for JACP: 1.57

Fig. 11. Examples of images for which the proposed method provides best compression ratios

CR for PAQ8: 1.18, CR for JACP: 1.10 | CR for PAQ8: 1.20, CR for JACP: 1.11 | CR for PAQ8: 1.22, CR for JACP: 1.12

Fig. 12. Examples of images for which the proposed method falls behind on PAQ8

Examples of images where PAQ8 outperforms JACP are shown on the Fig. 12. All these images contain highly textured regions.

Conclusions

In the paper we propose new fast and effective method JACP of additional lossless compression of JPEG images. JACP has the following advantages.

JACP provides higher average compression ratio than other fast methods (STUFFIT, WINZIP).

For 25% of images JACP provides higher compression ratio than all considered methods.

Compressed BPG image can be used for fast preview of JPEG images in cloud based image storage services.

Any better than BPG lossy image compression method can be used instead of BPG without modification of the proposed compression algorithm.

Main drawback of the proposed method is its limited efficiency for additional compression of highly textured images with noise like textures and fine details.

References

[1] G.K. Wallace, "The JPEG Still Picture Compression Standard", Comm. of the ACM, Vol. 34, pp. 30-44, 1991.

[2] D. Taubman,"JPEG 2000: Image Compr. Fundamentals. Standards and Practice", M. Marcellin. – Boston: Kluwer, 2002. – 360 p.

[3] N. Ponomarenko, V. Lukin, K. Egiazarian, J. Astola, "ADCT: A new high quality DCT based coder for lossy image compression" in Proceedings of LNLA, pp. 33-40, 2008.

[4] BPG [Electronic resource]: BPG Image format, https://bellard.org/bpg

[5] Sullivan G J, Ohm J R, Han W J, et al, "Overview of the high efficiency video coding (HEVC) standard", IEEE Transactions on circuits and systems for video technology, 22(12), 2012, pp. 1649-1668.

[6] N. Ponomarenko, V. Lukin, K. Egiazarian, E. Delp, "Comparison of lossy compression technique performance for real life color photo images", Proceedings of Picture Coding Symposium. – USA. – 2009. – 4 p.

[7] M. Mahoney, Data Compression Explained [Electronic resource], http://mattmahoney.net/dc/dce.html

[8] Y. Gringeler, D. Lovato, "System and method for lossless compression of already compressed files", US Patent 7502514, 2005

[9] WINZIP [Electronic resource]: WINZIP homepage, https://www.winzip.com

[10] N. Ponomarenko, K. Egiazarian, V. Lukin, J. Astola, "Additional lossless compression of JPEG images", Proceedings of 4th Symposium on Image and Signal Processing and Analysis. – Zagreb, Croatia. – 2005. – P. 117-120.

[11] N. Ponomarenko, V. Lukin, K. Egiazarian, J. Astola, "DCT based high quality image compression", Scandinavian Conference on Image Analysis. - 2005. - P. 1177-1185..

[12] Ponomarenko, V. Lukin, K. Egiazarian, J. Astola, Fast recursive coding based on grouping of Symbols, Telecommunications and Radio Engineering, Vol. 68, No 20, 2009, pp. 1857-1863.

[13] N. Kozhemiakina, V. Lukin, N. Ponomarenko, J. Astola, K. Egiazarian, JPEG compression with recursive group coding, Proceedings of Image Processing: Algorithms and Systems XIV, 2016, 6 p.

[14] M. Ponomarenko, N. Gapon, V. Voronin, K. Egiazarian, "Blind estimation of white Gaussian noise variance in highly textured images", Image Processing: Algorithms and Systems XVI, 2018, 5p.

[15] K. Egiazarian, J. Astola, M. Helsingius, P. Kuosmanen, Adaptive denoising and lossy compression of images in transform domain, Journal of Electronic Imaging, vol. 8, 1999, pp. 233-245.