http://dergipark.gov.tr/tujes
Trakya University Journal of Engineering Sciences, 18(1): 31-41, 2017
ISSN 2147–0308
Araştırma Makalesi / Research Article

ENTROPY BASED ESTIMATION ALGORITHM USING SPLIT IMAGES TO INCREASE COMPRESSION RATIO

Emir ÖZTÜRK[1*], Altan MESUT[1]

[1] *Department of Computer Engineering, Trakya University, Edirne-TURKEY*

Abstract: Compressing image files after splitting them into certain number of parts can increase compression ratio. Acquired compression ratio can also be increased by compressing each part of the image using different algorithms, because each algorithm gives different compression ratios on different complexity values. In this study, statistical compression results and measured complexity values of split images are obtained, and an estimation algorithm based on these results is presented. Our algorithm splits images into 16 parts, compresses each part with different algorithm and joins the images after compression. Compression results show that using our estimation algorithm acquires higher compression ratios over whole image compression techniques with ratio of 5% on average and 25% on maximum.

Keywords: Estimation algorithm; image compression; image processing; image complexity

SIKIŞTIRMA ORANINI ARTTIRMAK İÇİN BÖLÜNMÜŞ RESİMLERİ KULLANAN ENTROPİ TABANLI BIR TAHMİN ALGORİTMASI

Özet: Resim dosyalarını belirli sayıda parçalara böldükten sonra sıkıştırma işlemi yapmak sıkıştırma oranını arttırabilmektedir. Elde edilen sıkıştırma oranı resmin her parçasının farklı bir algoritmayla sıkıştırılması ile daha da fazla arttırılabilmektedir. Her algoritma farklı karmaşıklık değerlerinde farklı sıkıştırma oranı sağlamaktadır. Bu çalışmada bölünmüş resimlerden sıkıştırma sonuçları istatistikleri ve ölçülen karmaşıklık değerleri elde edilmiş ve bu sonuçları kullanan bir tahmin algoritması önerilmiştir. Algoritmamız resimleri 16 parçaya böler, her parçayı farklı bir algoritmayla sıkıştırır ve bu parçaları en son aşamada birleştirir. Sıkıştırma sonuçlarından görüldüğü üzere algoritmamız resmi tek parça halinde sıkıştırma işlemine göre ortalama %5 ve maksimum %25 daha iyi sıkıştırma performansı sağlamıştır.

Anahtar Kelimeler: Tahmin algoritması; Görüntü sıkıştırma; Görüntü işleme; Görüntü karmaşıklığı

INTRODUCTION

With the evolution of digital photography and cameras, the amount and resolution of photos are increasing gradually. Therefore, there is a storage problem for large amount of images with high resolution. To reduce the size of the photos, lossless or lossy image compression methods can be used. JPEG (CCITT Rec, 1992), JPEG2000 (ISO/IEC 15444-1, 2004; Christopoulos et al., 2000), JPEG XR (ITU-T Rec, 2009) and PNG (ISO/IEC 15948:2004, 2004) can be given as examples to these methods.

JPEG, which is widely used since 1992, gains compression by eliminating high frequency values using DCT (discrete cosine transform) (Ahmed et al., 1974). However, using JPEG with high compression ratios results in blocking effect because of DCT. JPEG2000, which uses DWT (discrete wavelet transform) (Mallat, 1989) instead of DCT, provides better image quality. However, its slower encoding process limits its usage. Because slow encoders are not suitable especially for digital cameras, embedded systems, smartphones and other types of devices that have powerless processors. JPEG XR is proposed in 2009 as an alternative to JPEG2000. JPEG XR uses PCT with integer values with the help of Hadamard matrices. With an additional process, blocking effect could also be reduced.

General purpose lossless data compression algorithms like DEFLATE, LZMA, PPMd or Bzip2 can also be used to compress images. PNG is a lossless still image compression algorithm which is based on DEFLATE. Although lossless compression is not efficient for compressing photographs, it will be better to use a lossless method when compressing low complex images like diagrams, logos, etc.

Our previous work show that, compression ratio can be increased by splitting image into several parts and compressing these parts with the same algorithm individually (Öztürk, 2012). Using different compression algorithms on each part can also increase compression ratio.

In this study, statistical results from image properties are obtained by splitting image files into certain parts. Gain on compression ratio by compressing each part with different algorithm is examined. Finally, an estimation algorithm based on acquired statistical results is developed and performance of the estimation algorithm is measured. In second section, common algorithms used for getting statistical results are given. In third section, statistical method to obtain results is proposed. In fourth section an estimation algorithm is proposed and results of the algorithm is presented. Last section contains conclusions about developed estimation algorithm.

COMMON COMPRESSION ALGORITHMS USED FOR STATISTICAL ANALYSIS

LZMA (LZ77)

LZ77 is a dictionary based compression algorithm which was developed by Abraham Lempel and Jakob Ziv (Lempel and Ziv, 1977). The algorithm works by keeping a history window (known as: sliding window) of the most recently seen data and comparing the current data being encoded with the data in this window. If a matching repeated sequence is found, a reference to the position in the sliding window and the length of the match is encoded. The size of sliding window affects the compression ratio. LZSS is a slightly modified version of LZ77 that provides better compression ratio (Storer and Szymanski, 1982)

The Lempel–Ziv–Markov chain algorithm (LZMA) uses a dictionary compression scheme which is similar to the LZ77 algorithm. However, LZMA uses stream of bits which is encoded using an adaptive binary range coder instead of a generic byte-based model. Implementation of this model is as simple as byte-based model and it gives much better compression ratio. LZMA2 is a simple container format that can include both uncompressed data and LZMA data. LZMA2 supports arbitrarily scalable multithreaded compression and decompression and efficient compression of data which is partially incompressible (Wikipedia, 2017).

DEFLATE & PNG

DEFLATE is a lossless compression algorithm, which was developed by Phil Katz in mid 90's (Deutsch, 1996). The compressed data is considered as set of blocks. Each block is compressed with using LZSS along with Huffman encoding. Huffman tree for each block is independent from previous and next block. Size of the compressible blocks are variable. If the encoder decided that Huffman tree is too big to encode data efficiently, the current block will be ended and a new block will be created. Huffman trees are added to encoded blocks before compressed data and these trees are also encoded with Huffman encoding. To obtain efficient compression ratio, minimum 3 repeated characters will be encoded. 256 different repetition count between 3 and 258 could be represented with one byte. Search buffer which is 32.768 bytes long will be represented with 15 bits and 1 bit is used for uncompressed data flag. If a repetition does not found in sliding window, actual byte values will be encoded with the same Huffman tree. However, the distance information will be encoded with a different Huffman tree (Feldspar, 2011; RFC 1951, 1951).

PNG (Portable Network Graphics) is a lossless image compression method that is based on DEFLATE (…). While GIF format is limited to 8-bit indexed color (256 color), PNG gives a much wider range of color depths and supports alpha channel transparency. Although PNG does not support animation intrinsically like GIF, it is now the most widely used lossless image compression method on the Internet (Gelbmann, 2013).

PPMd (PPM)

PPM (Prediction by Partial Matching) algorithm was published in 1984 by Cleary and Witten (Cleary and Witten, 1984). It tries to predict the next character using some previous encoded characters. In 90s, different variations of PPM showed up. The most used version of PPM was developed by Dmitry Shkarin and known as PPMd (Shkarin, 2002).

Based on the n-grams obtained from the previously encoded part of the input being compressed, the probability distributions of all the characters following these n-grams are stored. Although these probability distributions are compressed with Huffman or Arithmetic encoder, if $n > 5$ the required memory size will be too large. Generally arithmetic coding is preferred because possibility range could be very wide. If no prediction can be made based on previously encoded n symbols (the currently encoded symbol is not found in the n^{th} context), the algorithm encodes the probability of an escape character and search the symbol in $(n - 1)^{th}$ context. This process is repeated until a match is found. If no more symbols remain in context a fixed prediction is made.

Different approaches have been developed to determine the probability of the escape character. PPMd is one of them, which increments the count of the escape character every time it is used. In other words, PPMd estimates the probability of a new symbol as the ratio of the number of unique symbols to the total number of symbols observed.

Although PPM-based compressors often have a high compression ratio, they cannot be used to speed up network traffic due to their slow compression and decompression times.

Bzip2 (BWCA & BWT)

Bzip2 is a free and open-source file compression utility that uses the BWT (Burrows–Wheeler Transform). Although it has a lower compression ratio than PPM-based algorithms, it is widely used because it can perform faster compression and decompression.

BWT (Burrows and Wheeler, 1994) is a block sorting method which rearranges a character string into runs of similar characters. Generally, another transform like MTF (Move-To-Front coding) is used after BWT. BWT and MTF do not acquire compression but transform data into appropriate form to increase compression ratio of Huffman or Arithmetic encoding. Algorithms based on BWT which are developed lately use different transform techniques to increase compression ratio.

JPEG

JPEG is the most widely used still image compression standard since it was developed by Joint Photographic Experts Group in 1992. Proposed standard defines the compression and decompression stages but not the file format. JPEG is a lossy compression algorithm and the amount of compression could be specified using a quality factor. Lower quality factors provide higher compression ratios and lower image quality.

Compression stages of JPEG is as follows: Color space transformation, chroma subsampling, segmentation, discrete cosine transform, quantization, zig-zag scanning, run length encoding and entropy encoding.

Color space transformation from RGB to YCbCr must be done before downsampling. Downsampling will reduce the chrominance values (Cb and Cr) by averaging four neighboring pixels and storing only one (or two) value for each four pixels. This loss of information is not important since the eye is less sensitive to fine color details than to fine brightness details. After that, the image is split into blocks of 8×8 pixels and each block is transformed into frequency domain using DCT. According to the selected quality factor in the quantization stage, high-frequency components are stored with a lower accuracy than the low-frequency components. Each block is scanned in a zigzag order to create an appropriate entry for run-length encoding (RLE). In the last stage that is entropy encoding, Huffman or Arithmetic coding is used.

JPEG2000

JPEG2000 is another still image compression standard that was developed to overcome some of the disadvantages of JPEG and to achieve high quality images at high compression ratios. As we have mentioned before, it is a wavelet-based method. With using DWT, there will be blurring in the image at high compression ratios instead of blocking effect seen in JPEG compressed images. Compression stages in JPEG2000 can be simplified as follows: Color space transformation, segmentation (tiling), DWT, quantization, MQ encoding.

JPEG2000 has some additional features different from JPEG, like;

- Color space transformation can be done with Reversible Color Transform (RCT) that does not cause quantization errors (because it uses integers instead of floating point numbers).
- It can also perform lossless compression in the same architecture.
- The desired compression ratio can be given before compression. If the encoder predicts that it can compress the image to the specified size or smaller with lossless mode, it will use this mode.
- With the help of ROI (Reigon of Interest), partitions with important data could be compressed with high quality and other parts could be compressed with low quality.
- It supports alpha channel for transparency.
- It supports progressive transmission and multiple resolution representation with the help of its subband coding structure.
- It supports HDR (High Dynamic Range) by using floating point values for pixels.

JPEG XR

JPEG XR (JPEG extended range) is a newer standard and file format for photographic images based on Windows Media Photo (HD Photo). Its purpose is to obtain the speed of JPEG and quality of JPEG2000. If you want to compress an image that uses 16 bits for each channel with JPEG, each channel will lose 8 bits of information because JPEG can only use 8 bits per channel. JPEG XR is developed to support extended information for each channel (it supports bit depths of up to 32 bits). With the support of extended information, JPEG XR is optimal for compressing high resolution and HDR images.

Coding stages of JPEG XR are similar to JPEG's: Color space transformation, chroma subsampling, segmentation, prefiltering, Photo Core Transform (PCT), quantization, estimation of coefficients and entropy encoding.

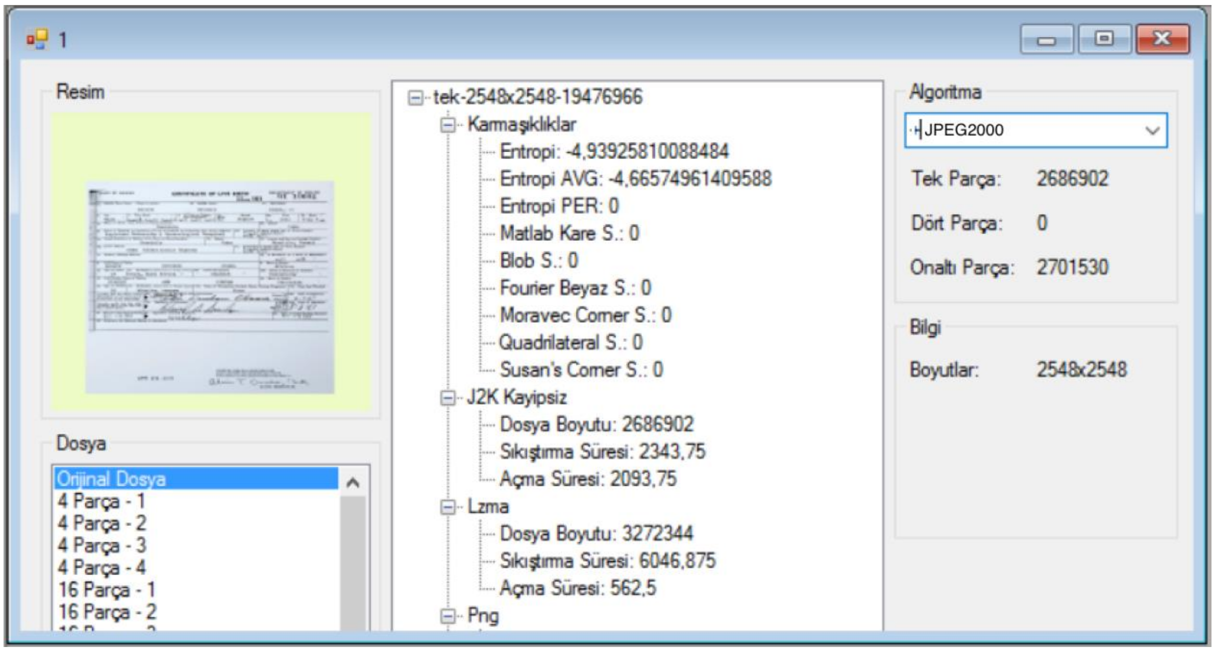
Some of the differences between JPEG and JPEG XR are:

- JPEG XR uses lossless color space transformation instead of lossy one in JPEG.
- JPEG XR uses 4 × 4 blocks instead of 8 × 8 (these blocks are grouped into 16 × 16 macroblocks).
- JPEG XR has an optional prefiltering step: Photo Overlap Transform (POT).
- JPEG XR uses PCT, which is a kind of 4 × 4 lossless DCT.
- Entropy coding in JPEG XR is more complex.
- JPEG XR supports lossless compression (like JPEG2000).

STATISTICAL METHOD TO ACQUIRE IMAGE PROPERTIES

Images can have different complexity in their different parts. After splitting image into parts, each part can be processed individually. Using the same algorithm with whole image and partitioned image can have different compression ratios. If each part is compressed with different algorithms, higher compression ratios can be obtained. Because, known compression algorithms will give different compression results on different complexity values (Öztürk and Mesut, 2016). PNG and general lossless compression algorithms obtain high compression ratio especially on low complexity images. Lossless modes of image compression algorithms i.e. JPEG2000 and JPEG XR obtain high compression ratio on high complexity images. For example, in a scanned document, the parts that contain photographs will have more complexity than the other parts that contain texts. Therefore, if we use a known general purpose lossless compression algorithm on low complexity parts instead of compressing them with JPEG2000 or JPEG XR, total compression ratio will be increased.

To obtain statistical results, 24bpp bitmap files were split into 4 and 16 equal parts. After that, 9 different compression algorithms (Deflate, Deflate64, LZMA, LZMA2, PPMd, Bzip2, JPEG2000, JPEG XR and PNG) are used on original image, 4 and 16 parts of the image. After splitting operation, compression ratios over 21 images are obtained. Different complexity values like entropy, different color count, quadrilateral square count are also obtained from original image and each part of the image. We developed an application for obtaining these results. Developed application is shown in Figure 1.


1

Resim

Dosya

Orijinal Dosya
4 Parça - 1
4 Parça - 2
4 Parça - 3
4 Parça - 4
16 Parça - 1
16 Parça - 2

```
tek-2548x2548-19476966
    Karmaşıklıklar
        Entropi: -4,93925810088484
        Entropi AVG: -4,66574961409588
        Entropi PER: 0
        Matlab Kare S.: 0
        Blob S.: 0
        Fourier Beyaz S.: 0
        Moravec Corner S.: 0
        Quadrilateral S.: 0
        Susan's Corner S.: 0
    J2K Kayipsiz
        Dosya Boyutu: 2686902
        Sıkıştırma Süresi: 2343,75
        Açma Süresi: 2093,75
    Lzma
        Dosya Boyutu: 3272344
        Sıkıştırma Süresi: 6046,875
        Açma Süresi: 562,5
    Png
```

Algoritma: JPEG2000

Tek Parça:	2686902
Dört Parça:	0
Onaltı Parça:	2701530

Bilgi

Boyutlar: 2548x2548

Figure 1. Application for Obtaining Image Properties

As seen in Figure 1, application allows selecting the whole image or its parts. Selected image or a part of it is shown on upper left. After selecting one of the images, main pane will show acquired complexity values and compression ratio of each known lossless compression algorithm. Compression results of the algorithms on whole image and split images could be shown for requested algorithm on the right.

Entropy is calculated on grayscale form of given image. Different color count is obtained for each color channel of the image and average of these values is used. Quadrilateral square count is calculated with splitting image until split part is under a given threshold value. Given image will be split into 4 equal parts and an integrity value for each part is calculated. If calculated integrity of the part is above the threshold value, the part will be split into 4 equal parts again.

Entropy value is directly proportional with complexity (entropy increases with increasing complexity). Analyzing complexity measures show that entropy is the most reliable criterion for complexity. Because of this, only entropy is selected for complexity measurement.

Comparing the compression results of 4 parts and 16 parts of images shows that dividing images into 16 parts gives better results than 4 parts. Therefore, splitting image into 16 parts is selected for proposed estimation algorithm.

The steps of statistical method based on acquired knowledge are given in Figure 2. As seen in Figure 2, the entropy value of whole image and parts of the image are acquired for complexity measurement.

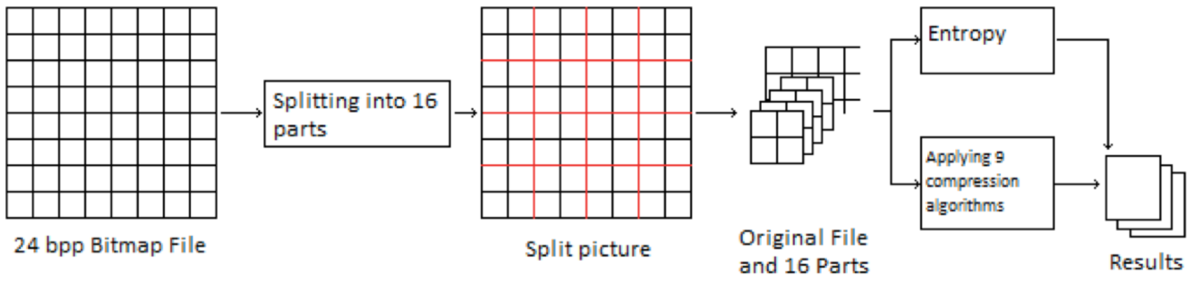


Figure 2. Steps of Obtaining Image Properties

ESTIMATION ALGORITHM

Using obtained statistical data, it can be identified which algorithm gives the most efficient results with given entropy value. Therefore, an estimation could be made to select which algorithm is suitable for compression without compressing the given part.

After analyzing the entropy values of given images, 2 algorithms out of 9 gives correlated results with entropy. Therefore, these 2 algorithms are selected for estimation to obtain proper results. Selected algorithms are LZMA and JPEG2000.

To obtain results for estimation, a corpus consists of 30 images larger than 2MP and 16 parts of these images (510 images in total) is used. Most of these images are scanned documents that have blank spaces, writings and pictures to show compression ratio of different algorithms on different level of complexity.

Figure 3 shows which algorithm is acquired best compression ratio over given entropy value. As seen in Figure 3, LZMA gives better results between 0.25 and 4.25 and JPEG2000 gives better results between 4.25 and 7.75. Especially when the entropy value is larger than 6, JPEG2000 gives better results on all images. Based on these results, an estimation algorithm could be proposed.

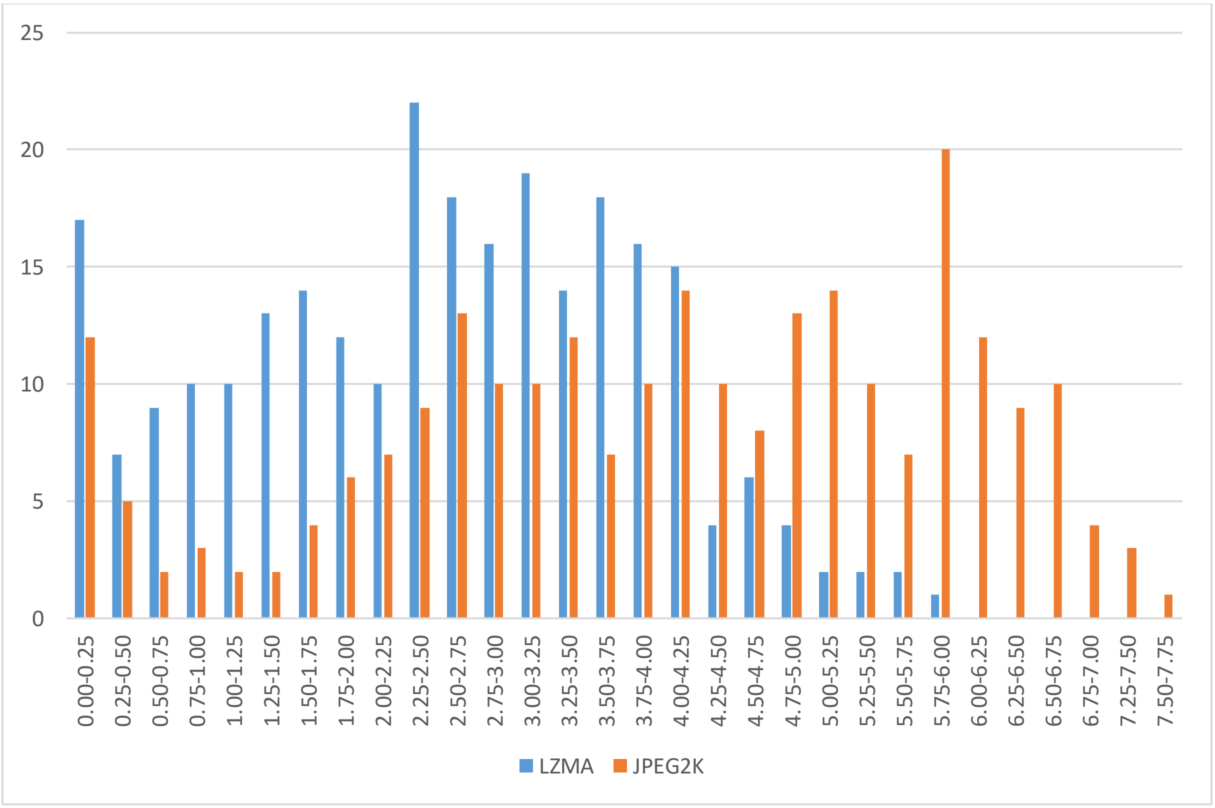


Figure 3. Best Compression Ratio Results at Given Entropy Values

Steps of the estimation algorithm are given in Figure 4. The algorithm splits given image into 16 parts. For each part, the algorithm first calculates the entropy value. Based on entropy value, the algorithm will choose a lossless compression algorithm and compresses that part with that algorithm. After compressing 16 parts with selected algorithms, the image will be joined.

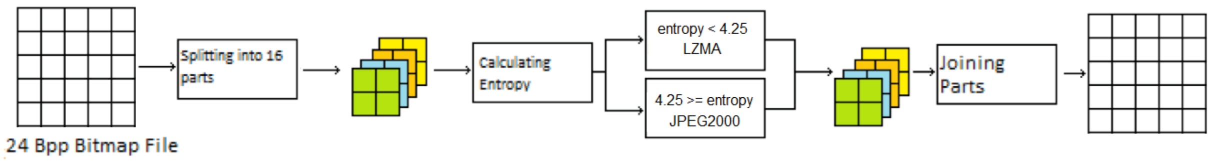


Figure 4. Steps of the Estimation Algorithm

An example of splitting operation is given in Figure 5. After splitting image into 16 parts, each part is compressed with most efficient compression algorithm. As seen on sample image, 6 parts are compressed with LZMA. Other 10 parts are compressed with JPEG2000. LZMA mostly gives better results on white spaces. JPEG2000 gives best results on parts with images or writings. Total size of the split image is smaller than compressing whole file with JPEG2000 or LZMA.

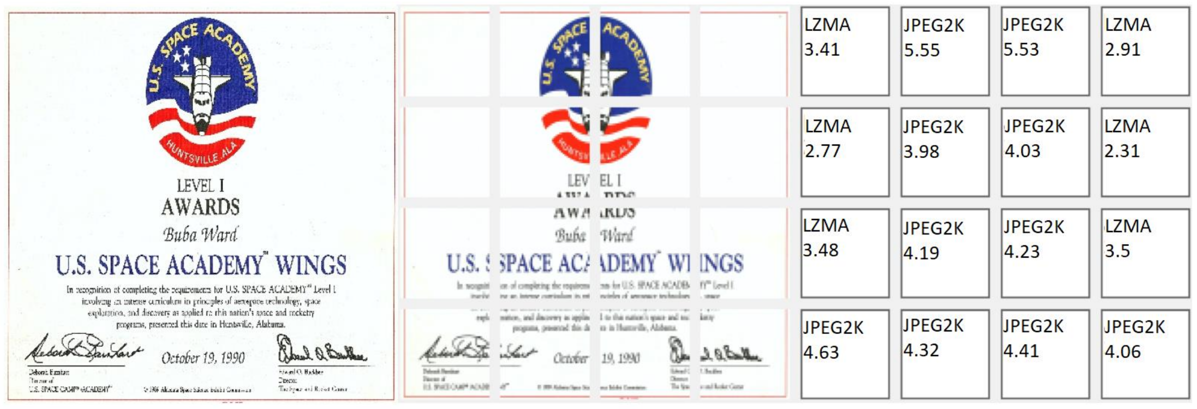


Figure 5. Splitting Operation of Sample Image

Compression ratios of our estimation algorithm and the best compression algorithm on whole image for 30 test images are given in Figure 6. Our estimation algorithm gives better results than compressing whole images with JPEG2000 or LZMA except 28th and 29th images.

Figure 7 shows compression gain of our estimation algorithm over whole image compression with the best algorithm. Most of these images acquired gain with using our estimation algorithm. Best gain is obtained on 23rd image file with the ratio of 24.24%.

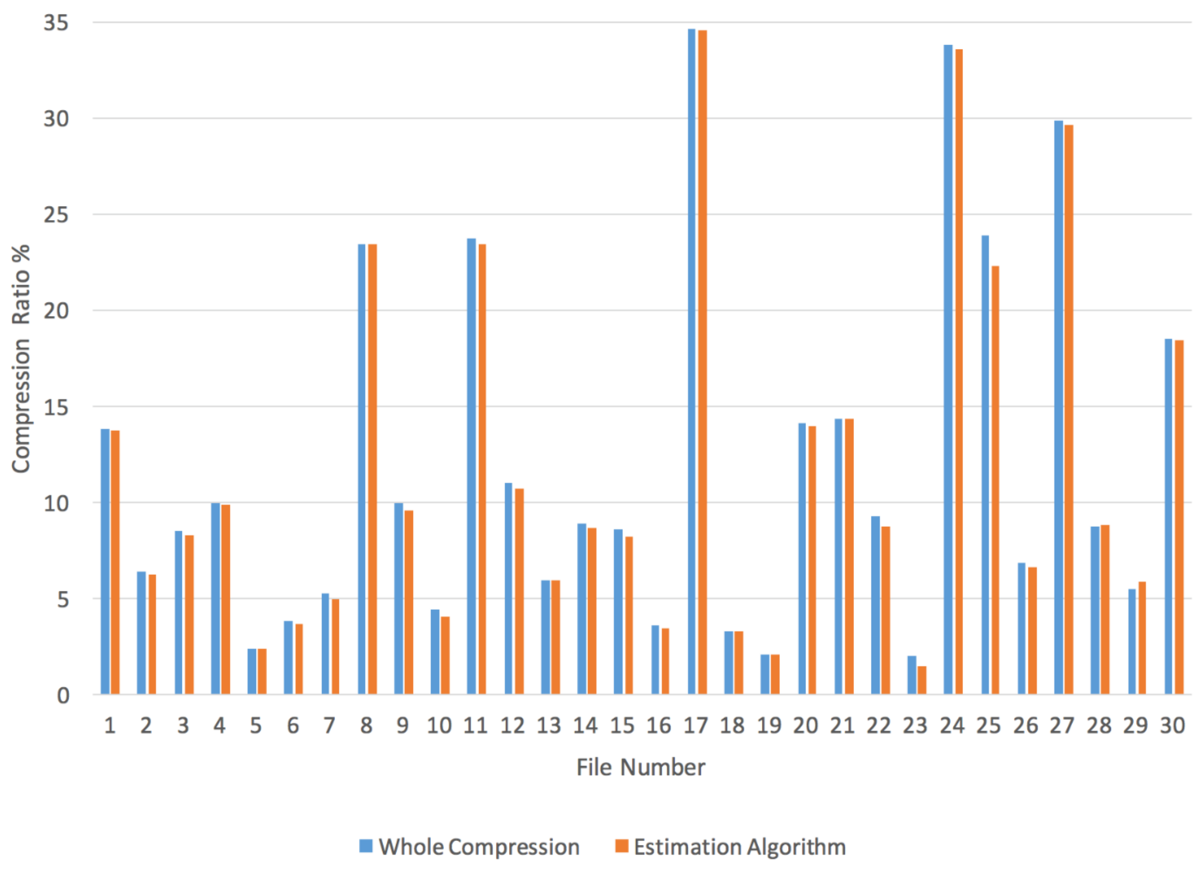


Figure 6. Compression Ratios of 30 Images

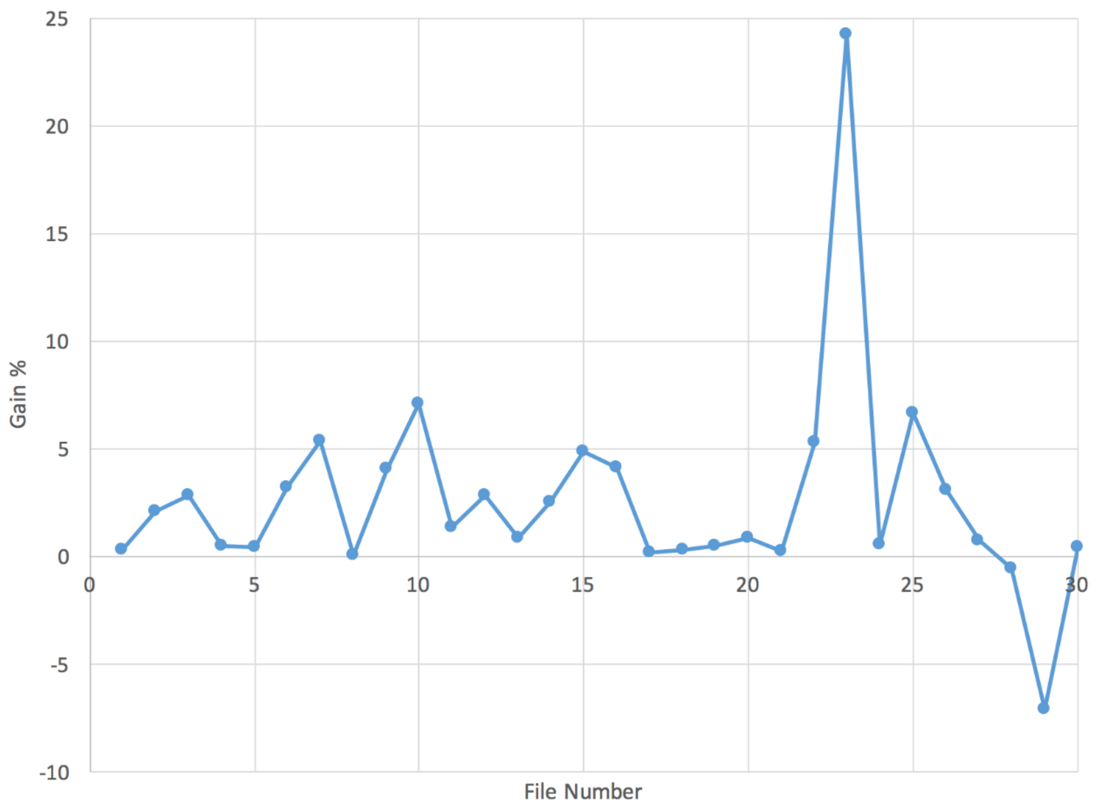


Figure 7. Obtained Compression Ratio Gain Over The Best Compression Algorithm

CONCLUSIONS

Splitting image into several parts is effective for increasing compression ratio. Besides, entropy is a decisive property for choosing the compression algorithm with best compression ratio. Within 9 algorithms, only two algorithms give stable results with the change of entropy. JPEG2000 is efficient mostly on images with high complexity and LZMA is efficient on low complexity images.

Although the gain obtained is small, gain will grow with image resolution. Success rate of the estimation algorithm is based on the amount and quality of statistical data. With the increase of amount and variety of data, algorithm will give more accurate results for different situations. Performance of estimation algorithm is also based on used known algorithms like JPEG2000 or LZMA.

With discovering different complexity methods, eliminated 7 algorithms could be also used in estimation algorithm. Using these algorithms with different situations could produce more efficient compression ratios.

An algorithm with lossy image compression algorithms could also be developed. When a user wants to specify the file size or amount of loss before compression, the algorithm could select proper lossy or lossless algorithm with right compression parameters to obtain desired result.

REFERENCES

1. CCITT Rec. T.81, (ISO/IEC 10918-1, 1994), Digital Compression and Coding of Continuous-Tone Still Images – Requirements And Guidelines (JPEG), 1992.
2. ISO/IEC 15444-1, JPEG 2000 image coding system: Core coding system. 2004.
3. CHRISTOPOULOS C, SKODRAS A, EBRAHIMI T. The JPEG2000 still image coding system: An Overview. *IEEE Transactions on Consumer Electronics*, 46(4), 1103-1127, 2000.
4. ITU-T Rec. T.832, (ISO/IEC 29199-2, 2010), JPEG XR image coding system: Image coding specification. 2009.
5. ISO/IEC 15948:2004, Information technology -- Computer graphics and image processing -- Portable Network Graphics (PNG): Functional specification, 2004.
6. AHMED N, NATARAJAN T, RAO KR. Discrete Cosine Transform. *IEEE Transactions on Computers*, 23(1): 90-93, 1974.
7. MALLAT S. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Pattern Analysis and Machine Intelligence*. 11(7), 674-693, 1989.
8. DEUTSCH LP. DEFLATE Compressed Data Format Specification version 1.3. *IETF Network Working Group, Request for Comments 1951*, 1996.
9. CLEARY J, WITTEN I. Data Compression Using Adaptive Coding and Partial String Matching, *IEEE Transactions on Communications*. 32 (4): 396–402, 1984.
10. SHKARIN D. PPM: One Step to Practicality, *Proceedings of IEEE Data Compression Conference*. 202-211, 2002, Utah, USA.
11. BURROWS M, WHEELER DJ. A block sorting lossless data compression algorithm, *Technical Report 124*, *Digital Equipment Corporation*. 1994.
12. ÖZTÜRK E. Transformation and Segmentation Operations on Images for Increasing the Effectiveness of Compression Methods, *MSc Thesis, Trakya University*, 2012.
13. FELDSPAR A. An explanation of the deflate algorithm. http://www.zlib.net/feldspar.html. Retrieved 2017-02-13.
14. ÖZTÜRK E, MESUT A. Finding the Optimal Lossless Compression Method for Images Using Machine Learning Algorithms, International Scientific Conference UNITECH'16, II,345-348 Gabrovo-Bulgaria, 2016.
15. ZIV J, LEMPEL A. A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory, Vol. 23, 1977, pp. 337-343.
16. ZIV J, LEMPEL A. Compression of Individual Sequences via Variable-Rate Coding. IEEE Transactions on Information Theory, Vol. 24, 1978, pp. 530-536.
17. WELCH TA. A Technique for High-Performance Data Compression. IEEE Computer, Vol. 17, 1984, No. 6, pp. 8-19.
18. STORER JA, Szymanski TG. Data compression via textual substitution. Journal of the ACM, Vol. 29, 1982, pp. 928-951.

19. Wikipedia. Lempel–Ziv–Markov chain algorithm. http://en.wikipedia.org/wiki/LZMA. Retrieved 2017-02-13.

20. Gelbmann M. The PNG image file format is now more popular than GIF. *W3Techs*. *Q-Success*. 2013-01-31.