First Attempt of Rapid Compression of 2D Images Based on Histograms Analysis

Danuta Jama
Institute of Mathematics
Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
Email: Danuta.Jama@polsl.pl

***Abstract*—Expanding technological advances and digitization generate more and more data, so the amount of space to store them is greater. To minimize the amount of stored data, efficient compression algorithms are needed. In this paper, the idea of the use of histograms for analysis of 2D images for the purpose of compression is shown. Performance tests were carried out, presented and discussed in terms of advantages and disadvantages.**

I. INTRODUCTION

Digital imagery has a huge impact on our lives. In every place on earth, at any time, a person has access to their data using a laptop or a cell phone connected to the Internet. And thus, the exchange of data between two points on the globe is no longer a problem. On the other hand, Internet has caused a wave of popularity of various social networking sites where people publish various photos of their life. This is just one of the phenomena of the last years, which intensified the amount of data and exchange information in the network at least several times.

These issues have caused several problems. First, the data traffic burden the entire network, so the upload speed or download can be reduced at the time. Secondly, the amount of data grows every day - files must be stored somewhere, so the number of servers will also increase. These are problems that can not be solved, but we can minimize them.

Data compression algorithms rely on conversion method of writing data in such a way that the weight of the data was smaller. In other words, the input file should be processed in order to save the file to a smaller number of bits.

Compression algorithms can be divided into two types - lossless and lossy. Lossless methods do not change the contents of the file, so the only form of writing a file is changed. The best-known algorithms is Huffman and Shannon-Fano coding, LZW or PNG. On the other hand, lossy algorithms operate by manipulating not only a record but also the quality of the file, eg .: DPCM or JPEG.

II. RELATED WORK

In recent years, data compression algorithms are experiencing a renaissance. The demand for methods of reducing the size of digital data is just one of the reasons for developing this subject. The second reason is the considerable development of related branches of computer science such as artificial intelligence methods that expand the possibilities of action of compression algorithms.

In 2012, Vikas Goyal [1] introduced the use of wavelet coding for the purpose of minimizing the size of image files. In addition, analysis of the use of different wavelets (including Haar or Dauchechies) EZW algorithm has been presented. Similarly, the authors of [2] presented an algorithm based on the combination of wavelet theory and chaos theory (fractals). For comparison, Cenugopal et al. [3] proposed the use of Arnold transform with chaos encoding technique showing the effectiveness of the creation of hybrid algorithms. In a similar time, Zhou et al. [4] presented the idea of encryption hybrid algorithm based on key-controlled measurement matrix.

The rapid development of artificial intelligence methods allowed to take a more random direction of compression, which can be seen on the example of the use of heuristic algorithms [5]–[7], or fuzzy logic [8], [9], what helped gain a significant advantage over other existing algorithms in terms of weight input files. And in [10], the authors introduced a novel predictor using causal block matching and 3D collaborative filtering.

Increasingly, modern computer science uses the latest achievements, but that does not mean that the older mathematical theories such as Fourier transforms are no longer used – in [11]–[14], a new approach to the use of either transform is described. Again in [15], Fracastoro et al. pointed to the use of transformation based on graphs.

In this paper, lossy compression algorithm for 2D images based on the analysis of the histogram is presented.

III. COMPRESSION ALGORITHM

To optimize the amount of compressed material from two-dimensional images, specific areas will be analyzed using the mechanism of the grid. The grid of size n pixels gets the area of the image and processes it. Then, the grid is shifted by n pixels and the operation is repeated until the entire image will not be covered with the grid.

The processing of the the grid is to calculate the histogram and check whether the histogram exceeds the threshold value γ (see Fig. 2, where a sample histogram is cut by the threshold

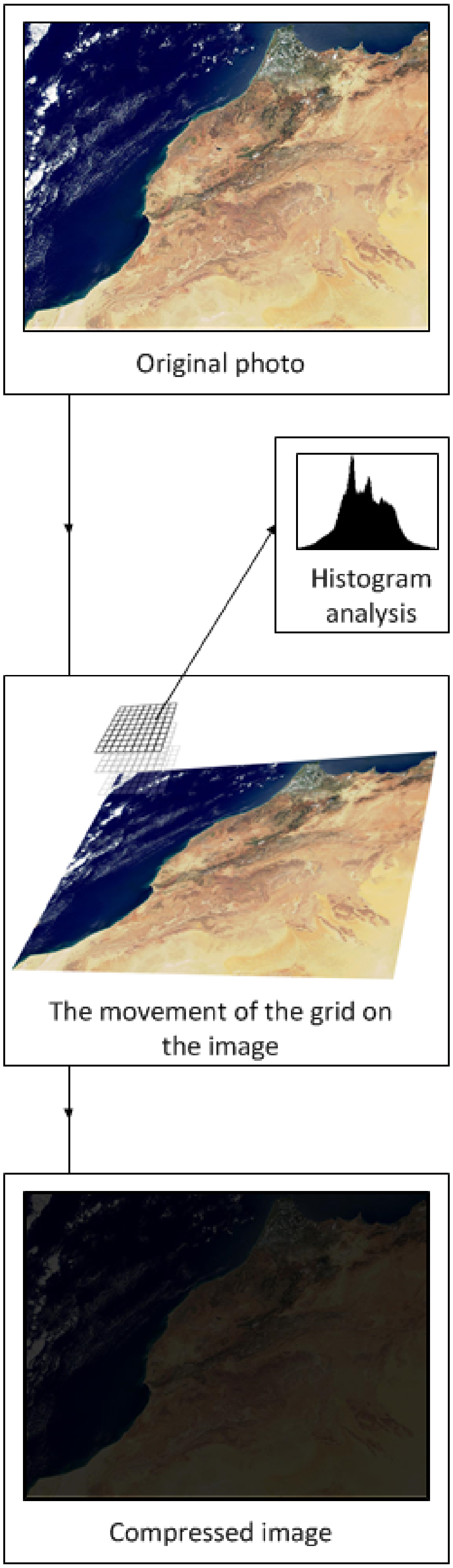


Fig. 1: A simplified model of the algorithm.

value). In the case where histogram of the area covered by a grid exceeds the threshold value, all the colors of pixels are scaled using the following formula

$$C_K = \begin{cases} \alpha K & \text{if} \quad \alpha K < 256 \\ 255 & \text{if} \quad \alpha K > 255 \end{cases}, \tag{1}$$

where K means a specific color component from the RGB model (R, G or B) and α is a given parameter in the range of $\langle 0, 2 \rangle$.

The parameter α manipulates the brightness of the image what can be represented as

$$\alpha = \begin{cases} \langle 0, 1) & \text{dim image} \\ 1 & \text{normal image} \\ (1, 2\rangle & \text{brighter image} \end{cases}. \tag{2}$$

The implementation of the described technique is presented in Algorithm 1.

Algorithm 1 The algorithm color change based on a histogram

```
Start
Load the image
Define the grid size n and a threshold value γ
Define an array representing the values of the histogram
max := 0
while grids does not cover the entire image do
   Load the grid on the current position
   for each pixel in the grid do
      Get the red value of a pixel to actualValue
      Increase the value of 1 in the array histogram at the
      position actualValue
      if max < actualValue then
         max = actualValue
      end if
   end for
   θ := false
   for each value ω in the histogram array do
      if ω ≥ γ then
         θ = true
      end if
   end for
   if θ == true then
      for each pixel in the grid do
         Recalculate color value using (1)
      end for
   end if
   Move the position of the value of n
end while
Stop
```

IV. Experiments

The algorithm has been tested for different sized images from the database *Earth Science World Image Bank* (available here – www.earthscienceworld.org). During the tests, the grid

The images were used only for research purposes and the author does not derive any financial benefit.

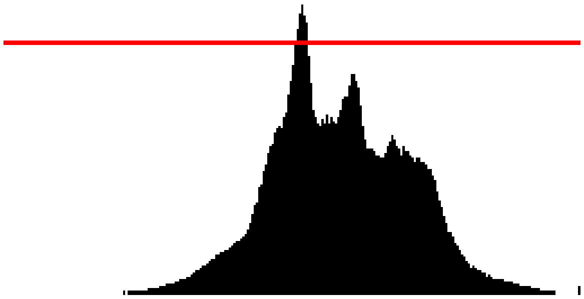


Fig. 2: The histogram with selected threshold value (red line).

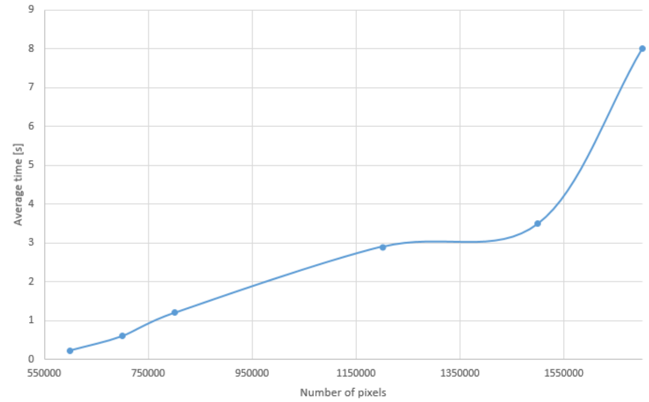


Fig. 4: The dependence of the average time from the size of the image.

size was set to 81 pixels, and thus 9×9 and a threshold value was set as 0.4.

Performed tests showed that for the small parameters α and γ, the image is darker but with a smaller size. The measurement results can be seen in Figures 5, 6 and 7. The weight of each file was measured before and after compression. Graphical comparison of the size is shown in Figure 3. The effectiveness of the proposed method was measured using the arithmetic mean. For some parameters, the compressed image has average 3.1 times lower weight. The dependence of the average running time of the algorithm from the size of the image is presented in Fig. 4. The image file is larger, the more time is needed to perform all operations. The time required for compression increases linearly for files composed of a maximum of 150 000 pixels. For larger files, the time increases fourfold.

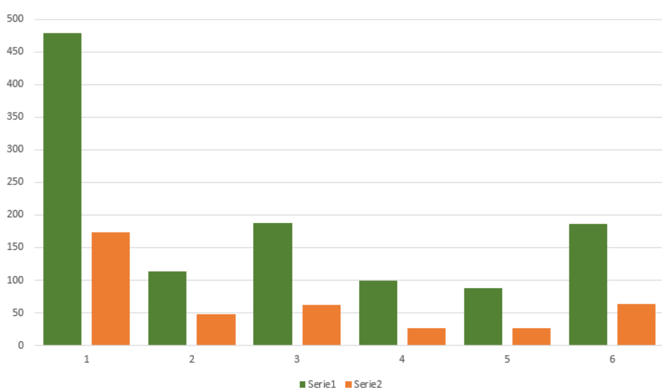


Fig. 3: Comparison of the size of files before and after compression.

V. Conclusion

The proposed method of compression of graphic images is able to reduce the image size by more than 50% with a good selection of threshold values. The biggest drawback of the algorithm is the large decrease in the visibility of the image by applying modification of colors, but the described technique allows to design an algorithm reverse if the input parameters are known.

In future work, the design of the algorithm reverse is planned. This method, lossy compression of the file and the ability to recover lost data would enable a much better flow of data on the Internet, because the two sides could have the original file, where the transferred file would be not only lower quality, but much smaller size.

References

[1] V. Goyal, "A performance and analysis of ezw encoder for image compression," *GESJ: Computer Science and Telecommunications*, no. 2, p. 34, 2012.

[2] A. Al-Fahoum and B. Harb, "A combined fractal and wavelet angiography image compression approach," *The Open Medical Imaging Journal*, vol. 7, pp. 9–18, 2013.

[3] D. Venugopal, M. Gunasekaran, and A. Sivanatharaja, "Secured color image compression and efficient reconstruction using arnold transform with chaos encoding technique," *signal*, vol. 4, no. 5, 2015.

[4] N. Zhou, A. Zhang, F. Zheng, and L. Gong, "Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing," *Optics & Laser Technology*, vol. 62, pp. 152–160, 2014.

[5] M.-H. Horng, "Vector quantization using the firefly algorithm for image compression," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1078–1091, 2012.

[6] R. Ramanathan, K. Kalaiarasi, and D. Prabha, "Improved wavelet based compression with adaptive lifting scheme using artificial bee colony algorithm," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 2, no. 4, pp. pp–1549, 2013.

[7] S. Yang, S. Wang, Z. Liu, M. Wang, and L. Jiao, "Improved bandelet with heuristic evolutionary optimization for image compression," *Engineering Applications of Artificial Intelligence*, vol. 31, pp. 27–34, 2014.

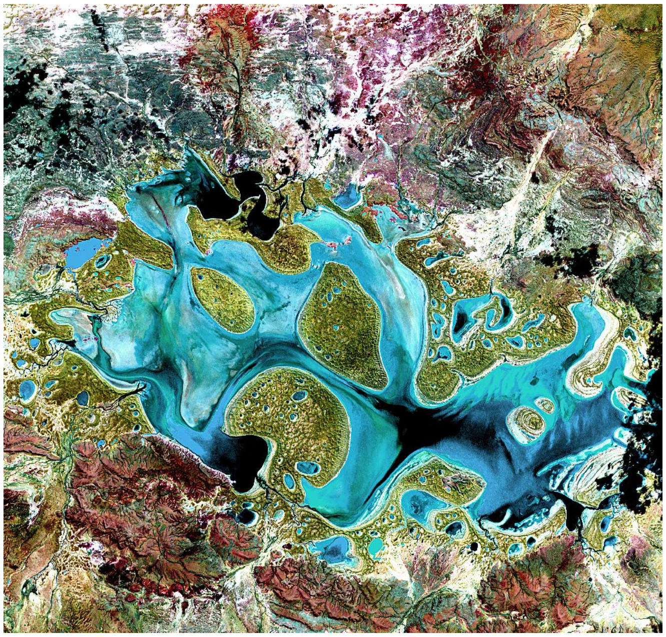
[8] V. S. Thakur and K. Thakur, "Design and implementation of a highly efficient gray image compression codec using fuzzy based soft hybrid jpeg standard," in *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on*. IEEE, 2014, pp. 484–489.

[9] D. M. Tsolakis and G. E. Tsekouras, "A fuzzy-soft competitive learning approach for grayscale image compression," in *Unsupervised Learning Algorithms*. Springer, 2016, pp. 385–404.

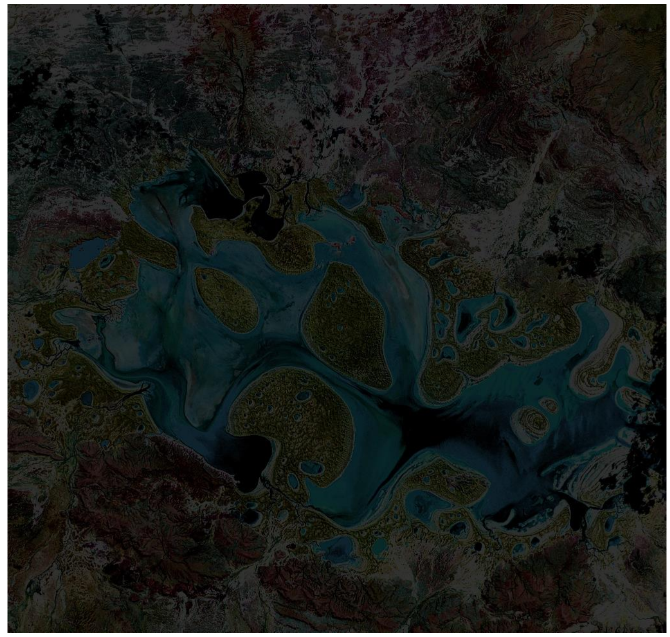
[10] R. Crandall and A. Bilgin, "Lossless image compression using causal block matching and 3d collaborative filtering," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5636–5640.

[11] M. Gupta and A. K. Garg, "Analysis of image compression algorithm using dct," *International Journal of Engineering Research and Applications*, vol. 2, no. 1, pp. 515–521, 2012.

[12] C. Rawat and S. Meher, "A hybrid image compression scheme using dct and fractal image compression." *Int. Arab J. Inf. Technol.*, vol. 10, no. 6, pp. 553–562, 2013.



(a) 479KB

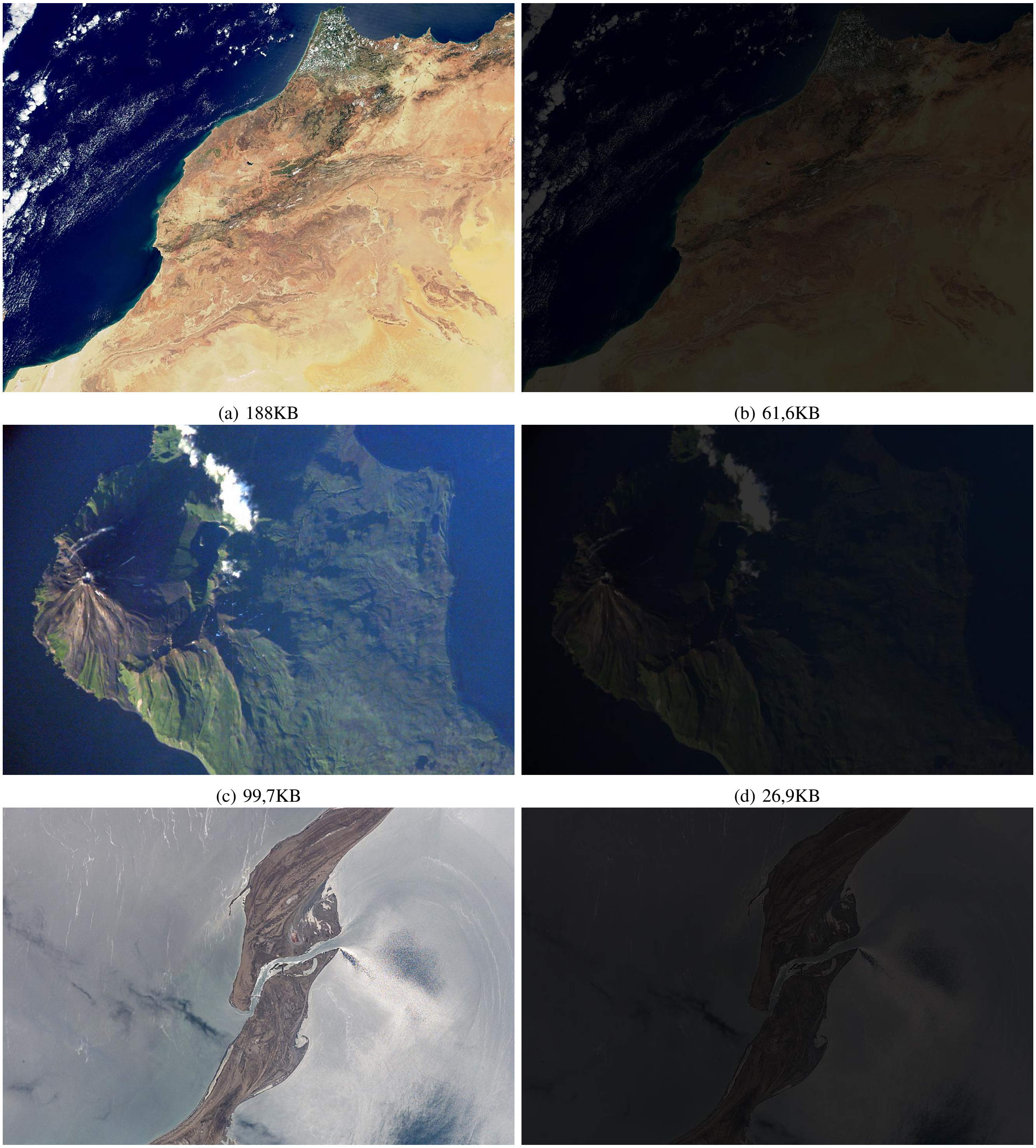


(b) 173KB

(c) 113KB

(d) 48,6KB

Fig. 5: Examples of quality compression.



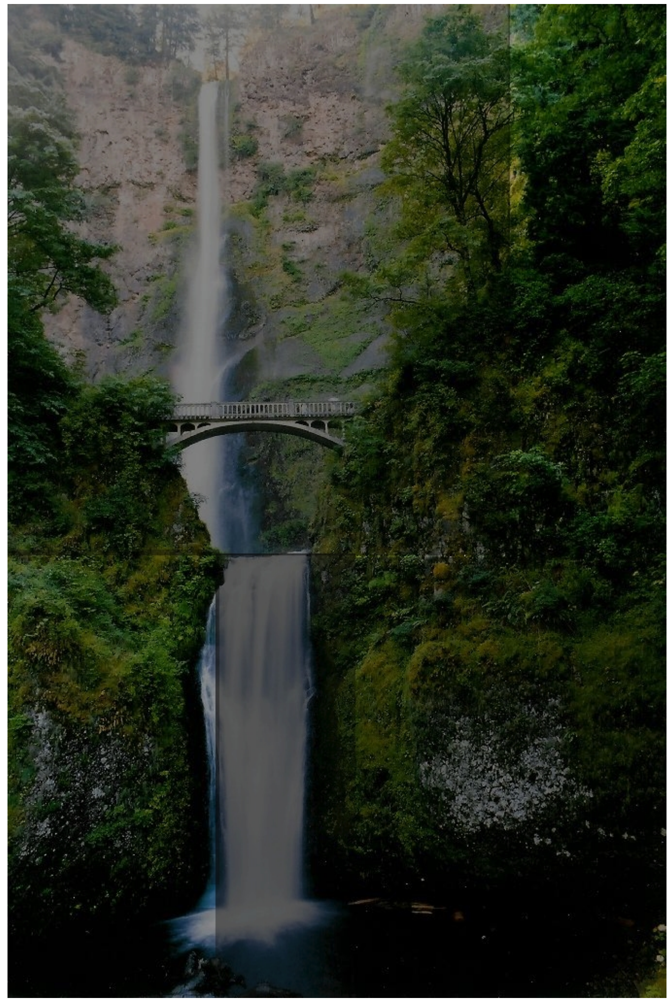
(a) 188KB (b) 61,6KB

(c) 99,7KB (d) 26,9KB

(e) 87,4KB (f) 26,3KB

Fig. 6: Examples of quality compression.

(a) 186KB



(b) 63KB

Fig. 7: Example of quality compression.

[13] T. Anitha and S. Ramachandran, "Novel algorithms for 2-d fft and its inverse for image compression," in *Signal Processing Image Processing & Pattern Recognition (ICSIPR), 2013 International Conference on*. IEEE, 2013, pp. 62–65.

[14] W. Hu, G. Cheung, A. Ortega, and O. C. Au, "Multiresolution graph fourier transform for compression of piecewise smooth images," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 419–433, 2015.

[15] G. Fracastoro, F. Verdoja, M. Grangetto, and E. Magli, "Superpixel-driven graph transform for image compression," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2631–2635.