Image Compression Based on the Visvalingam-Whyatt Algorithm

R. A. Melnyk[1], Y. L. Shokur[2]

[1]Prof. of software department, Lviv Polytechnic National University, S. Bandery Str., 12, Lviv, 79013, UKRAINE, E-mail: ramelnyk@polynet.lviv.ua,

[2]Master of software department, Lviv Polytechnic National University, S. Bandery Str., 12, Lviv, 79013, UKRAINE, E-mail:heisenbug94@gmal.com

Abstract – Algorithm of Visvalingam-Whyatt for polyline simplification (piecewise approximation) was used for image compression. The compression method was investigated for different tolerances and steps in the algorithm application for pixel rows and columns.

Key words – image compression, approximation, coding, simplification.

I. Introduction

Image compression - the process of reducing the size in bytes of graphics file without degrading the image quality to an unacceptable level [1]. Reducing file size allows to store more images in a fixed disk space. It also reduces the time required to transfer images via the Internet or to download web pages. The purpose of image compression is to reduce the redundancy for storage or transfer it in an effective manner.

Images can be compressed in several different ways. For example, for Internet two most common graphic formats exist: JPEG [2] and GIF [3,4]. JPEG method is often used for photos, while GIF format is commonly used for images in which geometric shapes are relatively simple.

Other methods of image compression include the use of fractals and wavelets. These methods are not widely used on the Internet. Implementation of these functions requires considerable computational costs. In this article a simplified approach is proposed, namely the polyline simplification (piecewise approximation). In most cases, these methods are used in cartography [5, 6], line simplification algorithms are often used to render high-resolution geographic features at an appropriate output resolution. They can be applied to both polylines and polygons.

II. Approximation of brightness functions

To obtain image function by color intensity colored image is converted to grayscale. Each pixel image may take values from 0-255, which is responsible for color from black to white.

For approximation of brightness functions, Visvalingam-Whyatt algorithm [5-7] was taken as basic.

1) Visvalingam-Whyatt algorithm.

Visvalingam-Whyatt algorithm [5-7] is a localized algorithm that works with a certain area and removes points based on the effective area. The effective area is defined as a square of the triangle formed by each point of the line and two other neighboring points.

While Douglas–Peucker [8,9] is the most well-known, Visvalingam's [7-9] algorithm may be more effective and has a remarkably intuitive explanation: it progressively removes points with the least-perceptible change. To determine which point removal incurs the smallest visible change, Visvalingam's [5-7] algorithm calculates the triangles formed by the successive triplets of points along each line; the point with the lowest bound of the triangle is removed. After each removal, the area of the adjacent triangles is recalculated, and the process repeats. This algorithm is also known as "Line Generalization by repeated elimination of points".

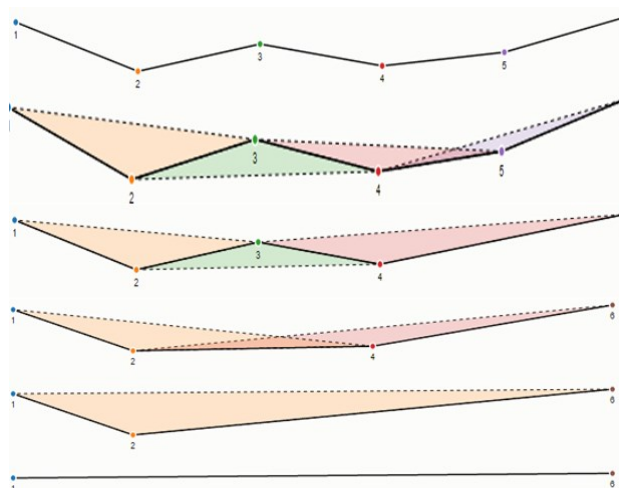The process of the curve approximation is shown on Fig. 1.

Fig. 1. Curve approximation
by Visvalingam-Whyatt algorithm

"The Algorithm:
1) *Excluding The first and last points, calculate the effective area of each point.*
2) *Remove the point with the least effective area (triangle) and place it into a stack.*
3) *Repeat steps 1 and 2 until all except the first and last points have been placed into the stack.*
4) *Starting at the top of the stack (the point with the most effective area) remove points and place them back into their proper position in the original point list*
5) *Repeat step 4 until the list is of the desired size."*[10].

Computational complexity of this algorithm is $O(N \log N)$.

The results of the brightness function approximation (width line and height line) using Visvalingam-Whyatt simplification algorithm for the test image for different values with the different tolerance values (0.5, 0.75, 1) are presented on Fig. 2, Fig.3, Fig.4, Fig.5, Fig.6, Fig.7..
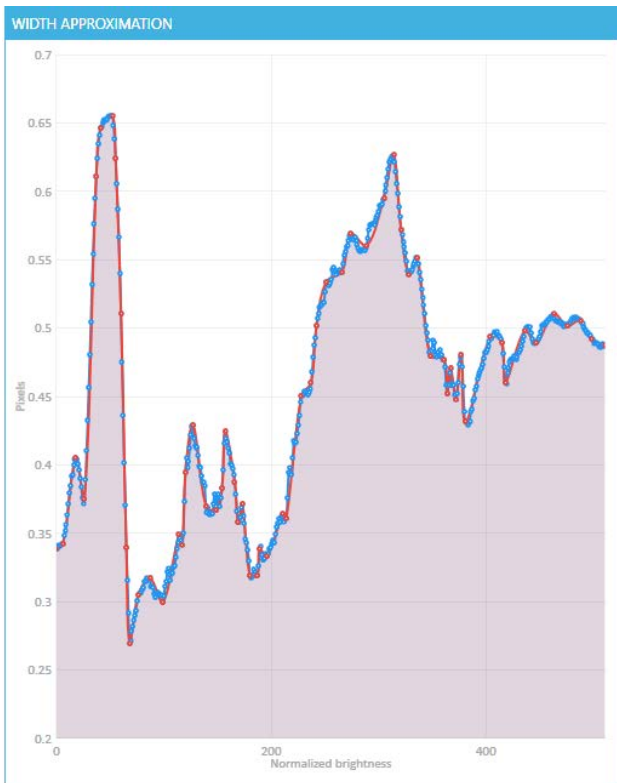
Fig. 2. Brightness function approximation (width) by Visvalingam-Whyatt algorithm (tolerance = 0.5)
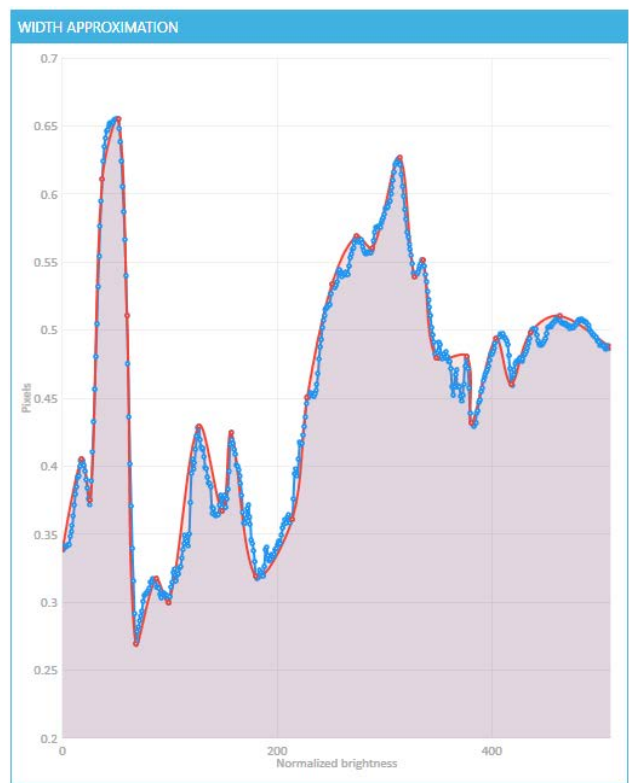

Fig. 4. Brightness function approximation (width) by Visvalingam-Whyatt algorithm (tolerance = 0.75)
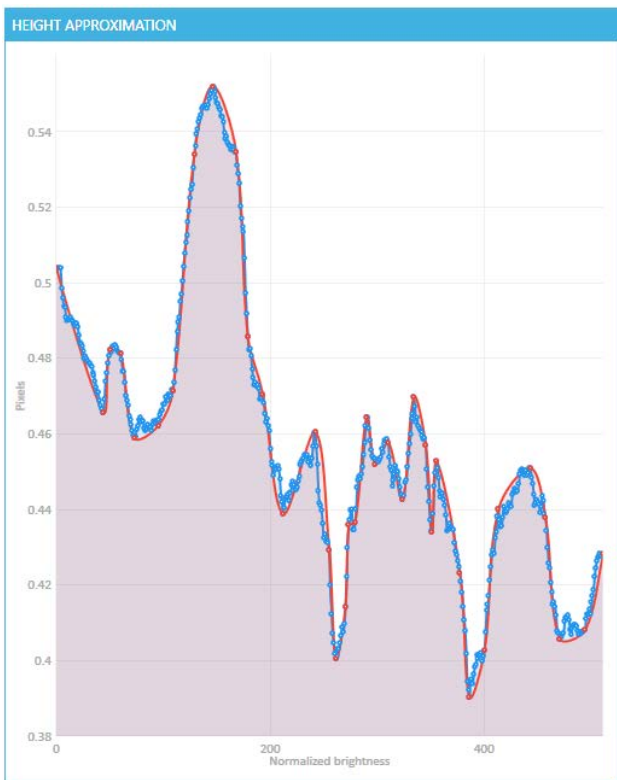

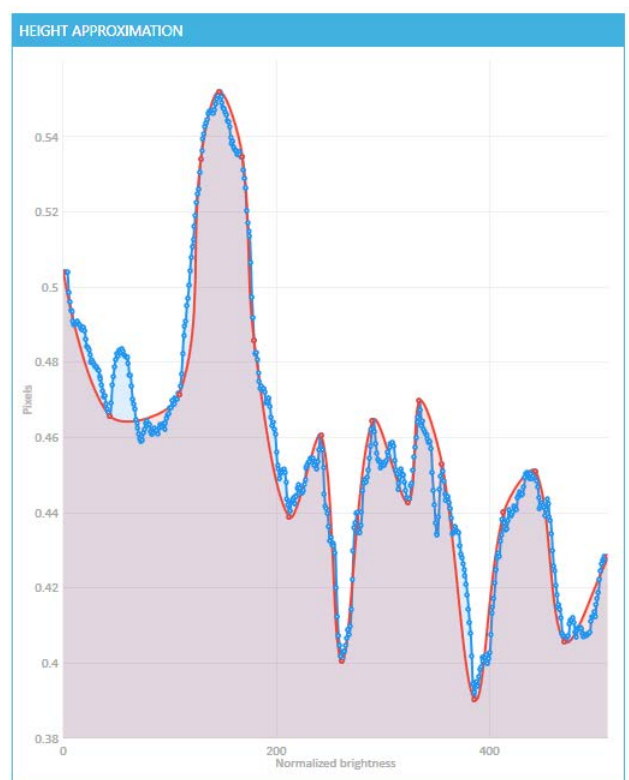Fig. 3. Brightness function approximation (height) by Visvalingam-Whyatt algorithm (tolerance = 0.5)

Fig. 5. Brightness function approximation (height) by Visvalingam-Whyatt algorithm (tolerance = 0.75)
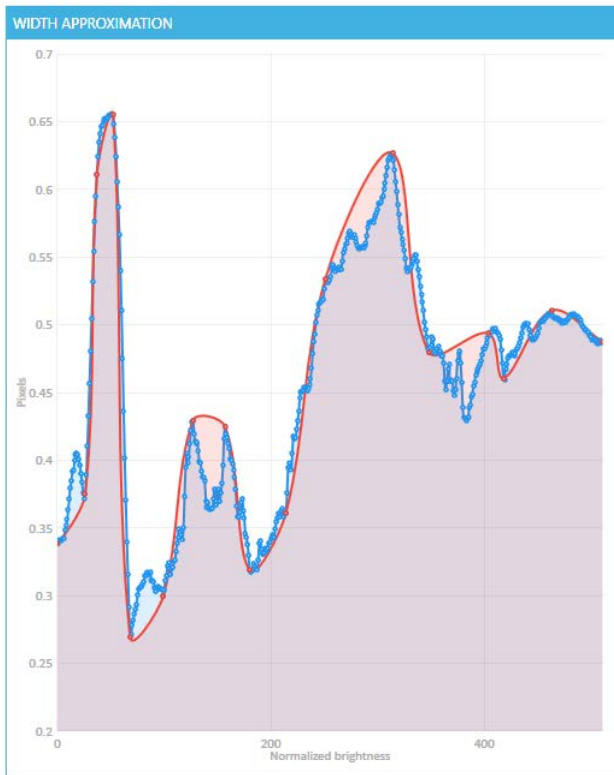
Fig. 6. Brightness function approximation (width) by Visvalingam-Whyatt algorithm (tolerance = 1)
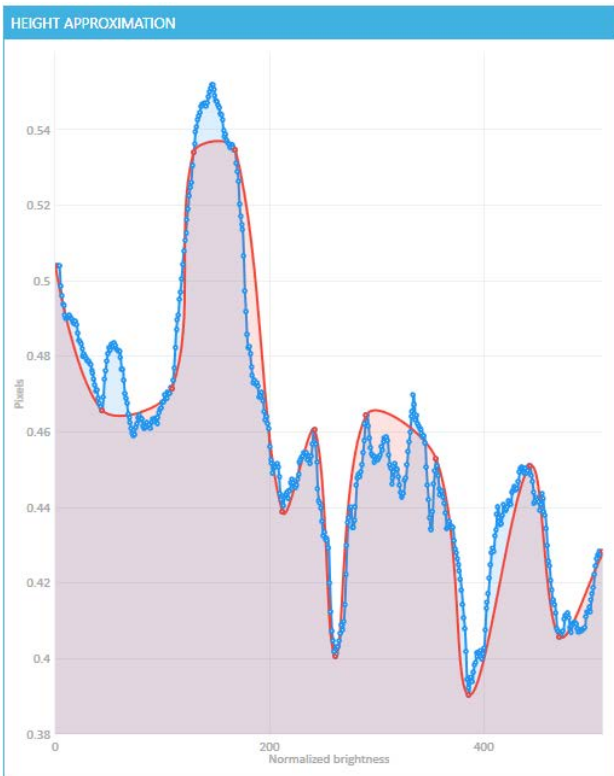

Fig. 7. Brightness function approximation (height) by Visvalingam-Whyatt algorithm (tolerance = 1)

III. Compression results

Comparison results using Visvalingam-Whyatt algorithm for different values of the maximum allowable deviation (tolerance) are shown in Table 1.

TABLE 1

IMAGE COMPRESSION USING VISVALINGAM-WHYATT ALGORITHM

	tolerance		
	0,5	0,75	1
"Lena" (512x512)			
(MSE), %	1,08	2,23	3,80
Compression ratio	1,3	3,34	6,54
Time, s	1,34	1,504	1,498

The Table 1 demonstrates increasing compression ratio and MSE according to the increasing tolerance value. The compression ratio is calculated by a reference on original image size that is a number of image pixels. We cannot compare these results with JPEG algorithm, because the last one is very complicated and consists of at least three powerful components; discrete cosine transformation, special type of matrix scanning and the Huffman algorithm for file compression. The given method has the same principles as the DCT algorithm. The last one neglects high frequencies by ignoring the DCT matrix coefficients. Our approximation does it when lines change some impulse curves [9].

The proposed algorithm was investigated by compression of the face image. The tolerance value was t=0.75, three compression experiments were conducted: one, two and three rows approximation by polylines simplification functions. Input image size is 512*512 pixels. In three cases the compression ratio was 6.84, 14. 07 and 21,47. The compressed images were characterized by corresponding MSE from original image: 1.09, 1.29 and 1.53. Then the compressed images were saved by jpeg-format. All this information presented in Table 2. In addition, this table shows sizes of the output (compressed) image after saving these images to jpeg format, and compressed coding array size: 38349, 18636 and 12207. Original image coding array size was 262144 byte.

After experiments with photos, research with the landscape image, which has larger size (1024*768) was conducted, and received results that are available in Table 3. MSE for three experiments are equal 4.1 for row skip = 1; 5.7 for row skip = 2 and 6.5 for row skip = 3. In three cases the compression ratio was 2.02, 4.55 and 7.39 and compressed coding array size: 388053, 172791 and 106311. Original image coding array size was 786432 byte.

Therefore, here we see possibility to improve a compression ratio by compressing a coding array itself, for example compressing by using ZIP, or another well-known methods.

IMAGE COMPRESSION BY DIFFERENT STEPS
USING VISVALINGAM-WHYATT ALGORITHM

Number of columns (step), t = 0.75			
original	1	2	3
(MSE), %	1,09	1,29	1,53
Compression ratio	6,84	14,07	21,47
Time, s	1,185	0,62	0,396
Coding array (b)	38349	18636	12207
jpg 28,6 kb	31,0 kb	28,3 kb	27,1 kb

TABLE 3

IMAGE COMPRESSION BY DIFFERENT STEPS USING VISVALINGAM-
WHYATT ALGORITHM FOR LANDSCAPE IMAGE

Number of columns (step), t = 0.75			
original	1	2	3
(MSE), %	4,1	5,7	6,5
Compression ratio	2,02	4,55	7,39
Time, s	7,36	3,75	2,54
Coding array (b)	388053	172791	106311
jpg 292 kb	254 kb	213 kb	197 kb

Conclusion

In this paper the image compressing method using polyline simplification was developed and investigated. In the researching, there were used different types of images (photos and landscapes) of different size, and with different steps and parameters (different tolerance, row skipping), with width or height approximation. For easy comparison, all results were presented in tables, and for the better clarity approximation results were shown on the diagrams.

Thus, consideed in this article method of image compression based on Visvalingam-Whyatt algorithm allows to fit the appropriate compression ratio with relatively small losses in image quality.

References

[1] Mahdi, O.A.; Mohammed, M.A.; Mohamed, A.J. "Implementing a Novel Approach and Convert Audio Compression to Text Coding via Hybrid Technique". International Journal of Computer Science Issues 9 (6, No. 3): 53–59. Retrieved 6 March 2013.

[2] A. Skodras, C. Christopoulos, T. Ebrahimi, "The JPEG 2000 Still Image Compression Standard", IEEE Signal processing magazine, 23 p., 2001.

[3] M. R. Nelson, "LZW data compression", Dr. Dobb's Journal, pp.28-36, 1989.

[4] Ziv, Inequalities and Algorithms for Universal Data Compression, Probl. Peredachi Inf., 1996, Volume 32, Issue 1, 35–40.

[5] Visvalingam, M. and Herbert, S., 1999, A computer science perspective on the bend simplification algorithm. Cartography and Geographic Information Science 26(4), 253-270.

[6] Visvalingam, M. and Whyatt, J.D., 1993, Line generalisation by repeated elimination of points. Cartographic Journal, 30(1), 46-51.

[7] Visvalingam, M. and Williamson, P.J, 1995, Simplification and generalization of large scale data for roads. Cartography and Geographic Information Science 22(4), 3-15.

[8] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves", Computer Graphics and Image Processing, 1(3): pp.244–256, 1972.

[9] D. Douglas, T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", The Canadian Cartographer, 10(2): pp.112–122, 1973.

[10] R. Melnyk, P. Viazovskyy, "Image compression using piecewise-linear approximation", Computer science and information technologies; 20-24 November, 2012 Lviv, Ukraine.

[11] SUNY Institute of Technology 2012. Available: http://web.cs.sunyit.edu/~poissad/projects/Curve/about_algorithms/whyatt.php