Comparing Freeman Chain Code 4 Adjacency Algorithm and LZMA Algorithm in Binary Image Compression

Resdiansyah¹, Justin Darmawan², Adrian Hartanto Wijaya³, Lukman Hakim⁴, Hendy Tannady⁵

¹Universitas Pembangunan Jaya, Banten, Indonesia 15413
²,³,⁴Computer Science Department, Universitas Bunda Mulia, Jakarta, Indonesia 14430
⁵Universitas Pembangunan Jaya, Banten, Indonesia 15413
*hendytannady@gmail.com

Abstract. Image is one product that has experienced a significant increase in quality. Insufficient internet capacity makes it difficult to access large images. Image compression is one way to solve this problem, but image compression can make image quality degrade. This research uses lossless compression algorithms, namely Chain Code Algorithm and LZMA Algorithm, to reduce the quality of the image and can be adjusted the tolerance value manually. In the application that will be created, the user can select the image to be compressed, then compress it according to the tolerance value that has been set and with the compression result of a text file, where the text file can be extracted and reconstructed into an image. There will be a preview of the results of the image compression and a description of the size and quality reduction levels. The conclusion of this research is that Chain Code Algorithm and LZMA Algorithm are able to compress image file sizes / images with extensions .jpg, .png, .bmp with a success rate of 90% from 50 tests.

Keywords: image, image compression, lossless compression, chain code algorithm, LZMA algorithm

1. Introduction

Informatics is one of the fields of science and technology that is developing rapidly so that it can play a role in helping humans obtain a more perfect life [1]. Since the discovery of a tool to capture an image in a two-dimensional plane (image) in the form of a camera, with the development of technology at this time, it does not only focus on the tools used to capture the image. The problem with images is the amount of storage space required because the image files obtained are very large and can occupy a lot of space in the storage media. In addition, large image data when sent over the network will make sending images from one place to another slow down [2]. A large bitmap image size will affect the process of transmitting or sending the image over a computer network. For example, a 24-bit bitmap image with dimensions of 1024 x 768 pixels requires about 2 Mb of storage memory. So that the larger the size of an image, the transmission speed is also slower and requires a large storage memory. Through compression with the right algorithm can change the image size with a slight change in image quality without losing the information in it [3].

Image compression aims to reduce data sets that are similar to images so that they can be stored in a smaller size or transmitted efficiently and to find image representations with reduced pixel correlation. The two basic

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd
principles used in image compression are redundancy and irrelevance. Redundancy is removing redundancy from the signal source and eliminating aberrations of pixel values that are not visible to the human eye [4]. In a communication system, the bandwidth required for digital image utilization is quite large. Because the channel capacity is limited, it is necessary to develop image data compression techniques [5]. Image quality improvement is needed to experience noise when sending via transmission line, image is too light or dark, image is not sharp, blurred, and so on [6]. Digital image processing is an image processing technique that aims to improve image quality so that it can be easily interpreted by humans or computer machines, which can be in the form of photos or moving images, digital image processing has several advantages, namely cheap, fast, and not damaging to what is measured and able to identify physically products objectively [7]. Reducing the file size allows more images to be stored in a certain amount of disk or memory space. The first type of compression is called lossless compression, while the latter is called lossy compression [8]. Compression requires algorithms and some prominent compression algorithm techniques such as Huffman, Run Length Encoding, Arithmetic, and Entropy. Image compression is data compression to reduce the digital size of the image. Compressed image formats are GIF, PNG, JPEG and others [9]. In the Lossless compression technique, data compression, when decompressed, becomes a replica of the actual data. When binary data is compressed, it is necessary to reproduce properly when decompressed. The actual image likeness is sufficient for the most objective, as far as errors or problems between the actual image and the compressed image can be avoided. It is used for only a few specific applications simultaneously with rigid requirements such as medical imaging [10]. Lossy compression is used to reduce the size of the bitmap image. First, the original image goes through four stages, namely, decomposition, quantization, modeling and coding, and finally the compressed image. Through this process, the image loses some size from its original form while maintaining its original quality, but in a compressed form [11].

Chain codes are a popular representation method for form rasterization in various scientific and engineering disciplines, the most intuitive chain code proposed by Freeman in 1961 using 8 pixel connectivity [12]. Chaincodes are used to represent the shape boundaries of each object in the image by registering a list of edge points along the contour and specifying the direction of the contours on each edge in a list of numbers. The direction of each edge is quantified to be one of 4 or 8 directions. The resulting chain code is based on determining the starting point at the first edge in the contour list and moving clockwise around the contour, traversing from one pixel to the next boundary pixel [13].

LZMA is a type of lossless compression algorithm that uses a large size to generate a compression scheme. The basic idea is to use multiple data structures to generate different symbols in the original file and then use a range encoder to encode these repeating symbols. It is a variant of the LZ family which differs in terms of construction and size used for compression [14]. The LZMA algorithm uses delta encoder and sliding dictionary encoder together with the LZ77 dynamic coding concept. The output of the LZMA encoder is represented in the form of a tuple (offset, length, new symbol). Delta filters form the input data stream for effective compression by sliding windows. It stores or transmits data in the form of differences between sequential data. The output of the first byte of the encoding is the data stream itself. The next bytes are stored

![Figure 1. Encoder and Decoder LZMA](image)
as the difference between the current and the current bytes before. While the output of the delta decoder is to maintain the first data stream and the next byte is stored in addition between the current data and the previous byte. For real-time data that is constantly changing, delta encoding makes dictionary slide more efficient [15].

2. Research Methodology
The research begins by identifying problems that arise in system design. Literature is obtained by collecting various written sources such as journals, books, archives and relevant documents. The system development uses the Waterfall mechanism, so that each process is carried out regularly and does not overlap. The Needs Analysis Stage aims to identify the needs in designing or building an application system. The system design stage designs the interface of the system so that it is easy to use by the user. The coding stage uses the C# programming language. The final stage is Testing, this stage uses the Chain Code 4 algorithm and the LZMA algorithm. System design is needed to describe and determine the process flow that will be carried out in the system in the form of diagrams that describe how the system works and interacts with users starting from flowcharts, use cases, activity diagrams, and gantt charts.

![Flowchart of Freeman Chain Code 4 Adjacency](image)

In Figure 2, illustrates the process of the Freeman Chain Code 4 Adjacency algorithm when forming a chain code as follows, the binary array of the original image that you want to compress, the terms or conditions i < image width, j < image height, if pixel (i,j) = 0, then move sideways, if pixel (i+1,j) = 0, then form code 1. If not then move on to the next process, if pixel (i,j+1) = 0, then form code 2. If not then continue to the next process, if pixel (i-1,j) = 0, then form code 3. If not then continue to the next process, if pixel (i,j-1) = 0, then form code 4. If not then process over.
3. Discussion and Analysis

Implementation begins by using the Freeman Chain Code 4 Adjacency, this method is by calculating the distance of movement along a digital curve or successive pixel boundaries based on 4 connectivity directions, the direction of each movement is coded using the number scheme \[ i \mid i = 0,1,2,3 \] which is the multiple of the 90 degree counterclockwise angle of the positive x-axis. The next stage uses the LZMA method to compress data files with a high compression level, the result is a file that is much smaller in size than before it was compressed. The use of this method is by calculating the encoding and decoding processes. The results of these calculations produce the values used for the image compression process. When the user runs the program, there will be an initial display of the program that will be run by the user. In the program display, the 'Select Image' button will be used to enter a sketch image that will be processed by the user. After the user inserts the image, the user will convert the image into binary form. After being binned, the images that have been entered by the user will be chain coded and mapped into csv format, after which they will be compressed using LZMA. Then, the compressed file will be extracted by the user by pressing the "Extract" button and to save by pressing the "Save" button. In the binarization process, it contains the process for converting an RGB image into a binary image, after which it is displayed in the picture box and the size of the binary image on the label. In the compression process, it contains the process to chain code the binary image, enter the coordinates and chain code in a new file with the .csv format, then perform LZMA compression on the csv file, display the compressed image results and the size of each file and path location. In the extraction process, contains the process for extracting the LZMA compressed file to the chain code file. After getting the chain code file, perform chain extraction to remap the binary image. Figure 3 shows a display of the extraction results.

![Figure 3. Display of Extraction](image)

Testing of application and algorithms - Tests conducted on 50 images with 3 different formats (.jpg, .png, .bmp) and with different sizes. The test results from the three different formats will be displayed in 5 pieces per format. In testing with the .jpg file format with various dimensions and different file sizes then the compression process was carried out using Freeman Chain Code 4 Adjacency and LZMA, it can be seen that all experiments resulted in successful conclusions with an average percentage of 82.35%. In testing with the .png file format, the original image inserted and converted into binary can be seen. Then five examples are .png files with various dimensions and different file sizes then the compression process is carried out using Freeman Chain Code 4 Adjacency and LZMA. The conclusion is that there is one failure in compressing the .png file format. Testing with the .bmp file format also shows that all tests were successful. Table 1 shows experiments on files with the .bitmap format.

| No | Original Image | Binary Image | Dimension (.ext) | Binary Size | Chain Size | LZMA Size | Percentage |
|----|----------------|--------------|------------------|-------------|------------|-----------|------------|
| 1  | ![Image](image) | ![Image](image) | 640x330 (.bmp)   | 633,654 bytes | 192,858 bytes | 14,347 bytes | 97.74%     |
The success rate of compression - the success rate of using the Freeman Chain Code 4 Adjacency and LZMA methods in performing compression is 90%. The number of successful attempts of the .jpg format file is 20 out of 20 data, the number of successful files with the .png format is 15 out of 20 data, while the file with the .bmp format is 10 out of 10 data. At the percentage level of compression, 45 test image data were successfully compressed and 5 data were unsuccessful. The average percentage of file size reduction using the Freeman Chain Code 4 Adjacency and LZMA methods is 56.9292%.

4. Conclusion
By converting the image into a chain code collection can increase the compression percentage using the LZMA compression algorithm. In images with .png and .jpg extensions, the more black in the image, the smaller the compression that occurs. The large number of redundant chain codes should lead to more optimal LZMA compression. On the other hand, there is an image with .bmp extension, although the black color is not as much as the image with .png and .jpg extension, and has many different patterns, the compression that occurs is very large. Differences in image extension have an influence on the compression performed. The success rate of using the Freeman Chain Code 4 Adjacency and LZMA methods in compression is 90% and the average percentage rate of file size reduction using the Freeman Chain Code 4 Adjacency and LZMA methods is 56.9292%.

5. References
[1] C. T. Utari, “Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra,” J. TIMES., Vol. 5, No. 2, pp. 24–31, 2016.
[2] R. Krasmala, A. B. Purba, Dan U. T. Lenggana, “Kompresi Citra Dengan Menggabungkan Metode Discrete Cosine Transform (DCT) Dan Algoritma Huffman,” J.OIN., Vol. 2, No. 1, pp. 1–9, 2017.
[3] S. I. Murpratiwi, Dan I. M. O. Widyantara, “Pemilihan Algoritma Kompresi Optimal Untuk Citra Digital Bitmap,” Maj. Ilm. Teknol. Elek., Vol. 17, No. 1, pp. 94–101, 2018.
[4] I. G. N. J. Suryaningrat, G. D. Sanjaya, Dan N. L. G. P. Suwirmayanti, “Kompresi Citra Digital Menggunakan Metode Discrete Cosine Transform,” In Proc. SINTAK, 2018, pp. 38–44.
[5] B. D. Raharja, Dan P. Harsadi, “Implementasi Kompresi Citra Digital Dengan Mengatur Kualitas Citra Digital,” In Citee 2018, 2018, pp. 71–77.
[6] A. Sindar, Dan R. M. Sinaga, “Implementasi Teknik Thresholding Pada Segmentasi Citra Digital,” J. Man. Pen., Vol. 1, No. 2, pp.48-51, 2017.
[7] E. D. Madyatmadja, L. Hakim, H. Tannady, L. Harianto. “Determination Position and Distance of Store Based on Wifi Using K-Nearest Neighbor and Floyd Warshall Algorithm,” Technology Reports of Kansai University, Vol. 62, No. 5, pp. 2379–2389, 2020.
[8] E. Sonia Dan E. J. Singh, “A Review On Image Compression Technique,” Int. J. Eng. Comput. Sci., Vol. 4, No. 7, pp. 13341–13345, 2015.
[9] S. M. S. Hilles, Dan M. S. Shafii, “Image Compression And Encryption Technique: Review Paper,” Int. J. Data. Sci. Rsch., Vol. 1, No. 2, pp. 1–6, 2018.
[10] M. Singh, S. Kumar, S. S. Chouhan, Dan M. Shrivastava, “Various Image Compression Techniques: Lossy And Lossless,” Int. J. Comput. Appl., Vol. 142, No. 6, pp. 23–26, 2016.
[11] I. Charles, and I. C. Udousoro, “The Application Of Selective Image Compression Techniques,” J. Soft. Eng., Vol. 6, No. 4, pp. 116–120, 2018.
[12] B. Žalik, D. Mongus, and N. Lukač, “A Universal Chain Code Compression Method,” J. Vis. Commun. Image R., pp. 8–15, 2015.
[13] F. Baji, and M. Mocanu, “Chain Code Approach For Shape Based Image Retrieval,” Ind. J. Sci. Tech, Vol. 11, No. 3, pp. 1–17, 2018.
[14] Z. B. Tariq, N. Arshad, and M. Nabeel, “Enhanced LZMA And BZIP2 For Improved Energy Data Compression,” In 4th International Conference on Smart Cities and Green ICT Systems, 2015, pp. 256–263.
[15] P. P. M, and T. S. S, “Hardware Implementation Of Lossless LZMA Data Compression Algorithm,” Prog. In. Sci. Eng. Rsch. J., Vol. 2, No. 3/6, pp. 201–205, 2014.