BIOS : Jurnal Informatika dan Sains

SEAN INSTITUTE VOL : 1 NO: 01 TAHUN 2023 E-ISSN: XXXX-XXXX

DIGITAL IMAGE COMPRESSION USING RUN LENGTH ENCODING METHOD

Kristian Siregar
Program Studi Teknik Informatika, Universitas Budidarma
Email: kristiansiregar@gmail.com

Abstract

The application of Run Length Encoding (RLE) algorithm in image compression cannot always reduce the size of the image compression results. Giving a Run sign or the number of pixels that have repeated succession can certainly change the size of the image file to be smaller, but very different from the repetitive image pixels but not sequential or not at all will certainly give a large size change in the file compression Image compression files that use Algorithm RLE in applications that are often used by users in general on computers can read image matrices. Thus RLE has a special ability to reduce image files from the others if there is a composition of the value of repetitive image pixels. And to decompress RLE to digital images is also very simple because the file type .rle has information on the order of matrix values consisting of two parts, the odd order is the pixel value for the image and while the even order is the value of the number of repeaters in the previous odd pixel value.

Keyword : Compression, Encoding, Image

1. INTRODUCTION

The image is a digital representation of the object image, which can not be separated from human needs. In general, image representation requires a fairly large memory, especially color images. This certainly greatly affects the availability of space (space) and data processing, especially image data in the development of science, especially in the field of computer science. Another problem that often arises in image processing with large size is when the transmission process (sending images) through the communication media. This will certainly slow down the delivery time. For this reason, it is necessary to develop an application for image compression that aims to minimize memory. Several compression modes have been developed such as the Huffman, Shanon Fano, Zip, Elias Gamma, and others. The general principle used for image compression is to reduce duplication of data in the image that often occurs, so that the memory used to represent the image becomes less than the original image representation. In this Final Project, the writer uses the Run Length Encoding (RLE) method for image compression. This method is used to compress images that have groups of pixels of the same gray degree. Image compression using the Run Length Encoding method is done by making a series of value pairs (P, Q) for each pixel row, where the P-value represents the gray degree value, while the Q value represents the number of consecutive pixels that have the gray degree**.**

RLE compression method is to add the same repetition of bytes/characters in a row and displays only a character that is repeated accompanied by the value of the number of bytes/character repetition, whereas for bytes/characters where no repetition occurs, the character will not be compressed. Generally digital images are rectangular or square (in some imaging systems some are hexagon-shaped) which have a certain width and height. In general, image files in JPEG / JPG, GIF, TIFF, PNG, and BMP formats. It is important to know about the characteristics of image file formats, so that when activities related to image processing are carried out, they can do it appropriately and wisely. These types and sizes are usually expressed in terms of points or pixels so that the image size is always around. Each point has coordinates according to their position in the image. These coordinates are usually expressed in positive integers, which can start from 0 or 1 depending on the system used. Each point also has a value in the form of a digital number that represents the information represented by that point. Image Compression is a data compression application performed on digital images to reduce the redundancy of data

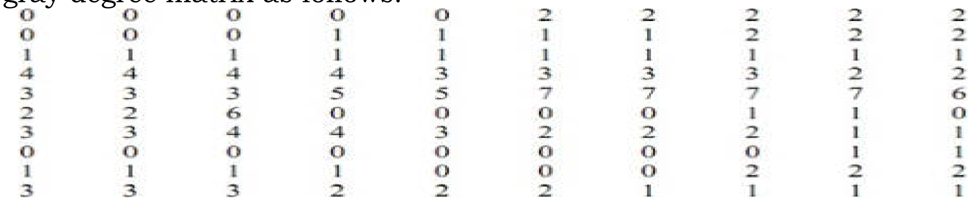
2. METHODS

Image

Image or Image is another term of image, which is visual information. The image has characteristics that are not owned by text data, namely the image is rich with information. There is a proverb that reads "a picture means more than a thousand words" a picture is more than a thousand words "That means of course a picture can provide more information than the information presented in the form of words. An image is an image in the two-dimensional plane (two dimensions) as shown in Figure 2.1. From a mathematical point of view, the image is a continuing function of the intensity of the light in the two-dimensional plane. The light source illuminates the object, the object reflects part of the beam. This reflection of light is captured by optical devices, such as the eye on humans, cameras, scanners, and so on, so that the shadow of the object called the image is recorded. [1]–[3]

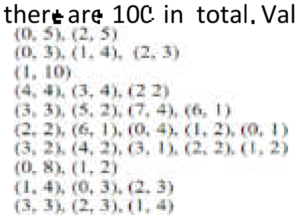
The RLE method is suitable for compressing images that have groups of pixels of the same gray level. Image compression by the RLE method is done by making a series of value pairs (p, q) for each pixel row, the first value (p) expresses the degree of gray, while the second value (q) states the number of sequential pixels that have the gray degree (called run-length). [4]–[6]

Review the 10 x10 pixel image with 8 degrees of gray which is expressed as a gray degree matrix as follows:



0	0	0	0	0	2	2	2	2	2
0	0	0	1	1	1	1	2	2	2
1	1	1	1	1	1	1	1	1	1
4	4	4	4	3	3	3	3	2	2
3	3	3	5	5	7	7	7	7	6
2	2	6	0	0	0	0	1	1	0
3	3	4	4	3	2	2	2	1	1
0	0	0	0	0	0	0	0	1	1
1	1	1	1	0	0	0	2	2	2
3	3	3	2	2	2	1	1	1	1

there are 100 in total. Value pairs for each run line generated by the RLE compression method:



(0, 5), (2, 5)
(0, 3), (1, 4), (2, 3)
(1, 10)
(4, 4), (3, 4), (2 2)
(3, 3), (5, 2), (7, 4), (6, 1)
(2, 2), (6, 1), (0, 4), (1, 2), (0, 1)
(3, 2), (4, 2), (3, 1), (2, 2), (1, 2)
(0, 8), (1, 2)
(1, 4), (0, 3), (2, 3)
(3, 3), (2, 3), (1, 4)

The size of the image before compression (1 degree of gray = 3 bits) is 100 × 3 bits = 300 bits, while the size of the image after compression (degree of gray = 3 bit, *run length* = = 4 bit): (31 × 3) + (31 × 4) bit = 217 bit

Compression Ratio = $(100\% - \frac{217}{300} \times 100\%) = 27.67\%$ which means 27.67% from

the original image has been compressed. Express as gray level values: 1 2 1 1 1 1 1 3 4 4 4 4 1 1 3 3 3 5 1 1 1 1 3 3

there are 24 values in total.

The pair of values from run generated by the RLE compression method: (1, 1) (2, 1) (1, 5) (3, 1) (4, 4) (1, 2) (3, 3) (5, 1) (1, 4) (3, 2)

Encoding results:

1 1 2 1 1 5 3 1 4 4 1 2 3 3 5 1 1 4 3 2

there are 20 values in all. So, we have saved 4 values.

The RLE method can be combined with the Huffman method to encode the RLE compression results in order to increase the compression ratio. First do RLE compression, then the results are compressed again using the Huffman method. [7]–[11]

3. RESULT AND DISCUSSION

Making pairs of pixels from the anak_muda30x30.bmp matrix by tracing the ruler of the matrix. In the same line, if it is known that there are the same values, sequentially combined and joined together by markers of the same number of values, and if not, they are marked with 1 only. And after finishing the first line continues to the next line until it is finished. There are 3 matrices from anak_muda30x30.bmp, so the three matrices are still visited in the same way.

Provisions in the pairing row are even sequencing values as the pair value. For example a piece of matrix R row 0 with its pair, for the complete part in Figure 3.6.
Row 0: 102 1 100 2 94 2 102 1 94 1 97 1 102 1 96 2 85 1 103 1 122 1 112 1 101 1 97 1 96 1 126 1 119 1
115 1 145 1 127 1 142 1 108 1 83 1 98 1 91 1 84 1 105 1 102 1

From the values above there are valuable pixels 102 (1), 100 (2), 94 (2), 102 (1), 94 (1), 97 (1), 102 (1), 96
(1), 85 (1), 103(1), 122 (1), 112 (1), 101 (1), 97 (1), 96 (1), 126 (1), 119 (1), 115 (1), 145 (1), 127 (1), 142 (1), 108 (1), 83 (1), 98 (1), 91 (1), 84 (1), 105 (1), dan 102 (1).

The 102 value actually exists 4 times in the line above but because it is not sequential the pair value is 1, while the value 100 is made 2 because in the Red matrix 2 times, and the value of 94 is 3 times but the sequential 2 others do not make it just 1 at the next 94 value. Stages in this storage is the formation into a compressed results file from the object image. From each table figure the values of R, G, and B are realized in the form of a file in which the definition is R, G, and B.

Table 1: Total value of pairing columns R, G, B

Red (R)	**Green (G)**	**Blue (B)**
Line 0 = 27x2 = 54 column	Line 0 = 28 x2 = 56 column	Line 0 = 27x2 = 54 column
Line 1 = 28 x2 = 56 column	Line 1 = 27x2 = 54 column	Line 1 = 25 x2 = 52 column
Line 2 = 29x2 = 58 column	Line 2 = 29x2 = 58 column	Line 2 = 27x2 = 54 column
Line 3 = 28 x2 = 56 column	Line 3 = 29x2 = 58 column	Line 3 = 27x2 = 54 column
Line 4 = 29x2 = 58 column	Line 4 = 29x2 = 58 column	Line 4 = 28x2 = 56 column
Line 5 = 28 x2 = 56 column	Line 5 = 25x2 = 50 column	Line 5 = 25x2 = 50 column
Line 6 = 28 x2 = 56 column	Line 6 = 28 x2 = 56 column	Line 6 = 29 x2 = 58 column
Line 7 = 28 x2 = 56 column	Line 7 = 29x2 = 58 column	Line 7 = 29 x2 = 58 column
Line 8 = 29x2 = 58 column	Line 8 = 29x2 = 58 column	Line 8 = 28 x2 = 56 column
Line 9 = 28 x2 = 56 column	Line 9 = 26 x2 = 52 column	Line 9 = 29x2 = 58 column

Line 10 = 28 x2 = 56 column	Line 10 = 29 x2 = 58 column	Line 10 = 29x2 = 58 column
Line 11 = 28 x2 = 56 column	Line 11 = 28 x2 = 56 column	Line 11 = 29x2 = 58 column
Line 12 = 28 x2 = 56 column	Line 12 = 28 x2 = 56 column	Line 12 = 28 x2 = 56 column
Line 13 = 29x2 = 58 column	Line 13 = 28 x2 = 56 column	Line 13 = 28x2 = 56 column
Line 14 = 28 x2 = 56 column	Line 14 = 29x2 = 58 column	Line 14 = 27 x2 = 54 column
Line 15 = 28 x2 = 56 column	Line 15 = 29x2 = 58 column	Line 15 = 29x2 = 58 column
Line 16 = 29x2 = 58 column	Line 16 = 28x2 = 56 column	Line 16 = 29x2 = 58 column
Line 17 = 29x2 = 58 column	Line 17 = 29x2 = 58 column	Line 17 = 29x2 = 58 column
Line 18 = 28 x2 = 56 column	Line 18 = 27 x2 = 54 column	Line 18 = 28 x2 = 56 column
Line 19 = 29x2 = 58 column	Line 19 = 29x2 = 58 column	Line 19 = 28 x2 = 56 column
Line 20 = 29x2 = 58 column	Line 20 = 28x2 = 56 column	Line 20 = 28 x2 = 56 column
Line 21 = 29x2 = 58 column	Line 21 = 29x2 = 58 column	Line 21 = 29x2 = 58 column
Line 22 = 28 x2 = 56 column	Line 22 = 29x2 = 58 column	Line 22 = 29x2 = 58 column
Line 23 = 29x2 = 58 column	Line 23 = 29x2 = 58 column	Line 23 = 28x2 = 54 column
Line 24 = 27x2 = 54 column	Line 24 = 26x2 = 52 column	Line 24 = 29x2 = 58 column
Line 25 = 28 x2 = 56 column	Line 25 = 28 x2 = 56 column	Line 25 = 28 x2 = 56 column
Line 26 = 29x2 = 58 column	Line 26 = 28x2 = 56 column	Line 26 = 29x2 = 58 column
Line 27 = 29x2 = 58 column	Line 27 = 27x2 = 54 column	Line 27 = 28 x2 = 56 column
Line 28 = 29x2 = 58 column	Line 28 = 29x2 = 58 column	Line 28 = 29x2 = 58 column
Line 29 = 29x2 = 58 column	Line 29 = 29x2 = 58 column	Line 29 = 29x2 = 58 column

From total results = total R + total G + total B = 1704 Bytes + 1690 Bytes + 1688 Bytes = 5082 Bytes = 4.96 KB Capacity swelling of 4.96 KB compared to the image of an object anak_muda30x30.bmp is only 2.74 KB, this is due to the large number of image pixels in the matrix which has a value of 1 in the sequence so that when copying the pairing it becomes 2 times the actual image size thus for the use of the Run Length Encoding algorithm in image compression if many values are not the same in a sequence then the result is not loading or reducing capacity.

4. CONCLUSION

To compress the pixel information obtained by giving a Run marker on each pixel visited. The Run Marker is the number of pixels visited worth the same amount repeatedly or once. As for calculating the ratio of the results of compression to the initial image, that is the ratio = 100% - (size of the result of compression/size of the initial image) * 100%.

REFERENCE

E. A. B. da Silva and G. V. Mendonca, "Digital Image Processing," in *The Electrical Engineering Handbook*, 2005.

S. Van Der Walt *et al.*, "Scikit-image: Image processing in python," *PeerJ*, 2014.

V. Hlavac, "Fundamentals of Image Processing," in *Optical and Digital Image Processing: Fundamentals and Applications*, 2011.

W. Li, B. Gao, S. M. Lee, K. Bennett, and D. Fang, "RLE-1, an E3 Ubiquitin Ligase, Regulates C. elegans Aging by Catalyzing DAF-16 Polyubiquitination," *Dev. Cell*, 2007.

B. Houston, M. B. Nielsen, C. Batty, O. Nilsson, and K. Museth, "Hierarchical RLE Level Set: A compact and versatile deformable surface representation," *ACM Trans. Graph.*, 2006.

K. Li, A. Hüsing, and R. Kaaks, "Lifestyle risk factors and residual life expectancy at age 40: A German cohort study," *BMC Med.*, 2014.

R. Krasmala, A. Budimansyah, and U. T. Lenggana, "Kompresi Citra Dengan Menggabungkan Metode Discrete Cosine Transform (DCT) dan Algoritma Huffman," *J. Online Inform.*, 2017.

A. Wibowo, "Kompresi data menggunakan metode huffman," *Semantik*, 2012.

T. R. Silviani and A. Arfiana, "Teknik Kompresi Citra Menggunakan Metode Huffman," in *SEMINAR NASIONAL MATEMATIKA DAN PENDIDIKAN MATEMATIKA UNY*, 2016.

Petrus Santoso, "Studi Kompresi Data dengan Metode Arithmetic Coding," *J. Tek. Elektro*, 2001.

D. Boedi, H. C. Rustamaji, and M. A. Nugraha, "Aplikasi Kompresi Sms Berbasis Java Me Dengan Metode Kompresi Lzw-Huffman," *Semin. Nas. Inform.*, 2009.