LOSSY IMAGE COMPRESSION USING FREEMAN CHAIN CODE REPRESENTATION

Sobiha Somasundaram[1], Venuja Niranjan[2] and Santhirakumar Pujinthan[1]

[1]University of Jaffna
[2]Department of ICT, Faculty of Technological Studies, University of Vavuniya

Abstract

In this study, we suggested an algorithm that made use of a Freeman chain code-based compression technique. The method consists of two segments such as compression and decompression. The first is a compression algorithm that begins by obtaining the chain code for a specific colour value and then saving the location of the start point for the chain code, colour value, level of the image, and chain code in a compressed text file. The next step is to remove all colour values associated with the chain code from the input image and shrink the input image. The algorithm repeats the previous steps until there are no colour values with significant chain code. The second step is to create the original image using the chain code, start point, colour value, and level of the image. The second part is to reconstruct the original image by using the start point, colour value, and chain code. We discover that this method is appropriate for the representation of lossy images based on the findings of our experiments. In comparison to the Joint Bilevel Image Experts Group (JBIG) compressor, the results are more effective at compressing data.
Keywords*: Lossy Image Compression, Freeman Chain Code, Decompression, Joint Bilevel Image Experts Group*

1 INTRODUCTION

Compression is practically always employed. The majority of modems utilize compression, HDTV uses MPEG-2 compression, and all images you download from the internet are compressed. Several file systems also automatically compress data when storing them. An encoding technique that takes a message and creates a ”compressed” representation of it (ideally with fewer bits) and a decoding algorithm that reconstructs the original message or a close approximation of it from the compressed form make up the two halves of compression.

In order to save storage space and reduce transmission bandwidth requirements, image data files are compressed by removing any redundancy that may be present in them. Lossless and lossy data compression algorithms can be described. Each stage of compression and decompression using a lossy approach degrades the quality of the images. Compared to lossless approaches, lossy techniques often offer much higher compression ratios. Lossy image compression techniques include vector quantization, JPEG, subband coding, and fractal-based coding, among others.

The huffman encoding, entropy encoding, arithmetic coding, run-length encoding and quadtree techniques are examples of lossless picture compression techniques. Data compression minimizes the number of bits by locating and removing statistical redundancy; as a result, the suggested system's compression may encode colour values even when they are not evenly distributed, in contrast to many previous compression techniques that do.

2 METHODOLOGY

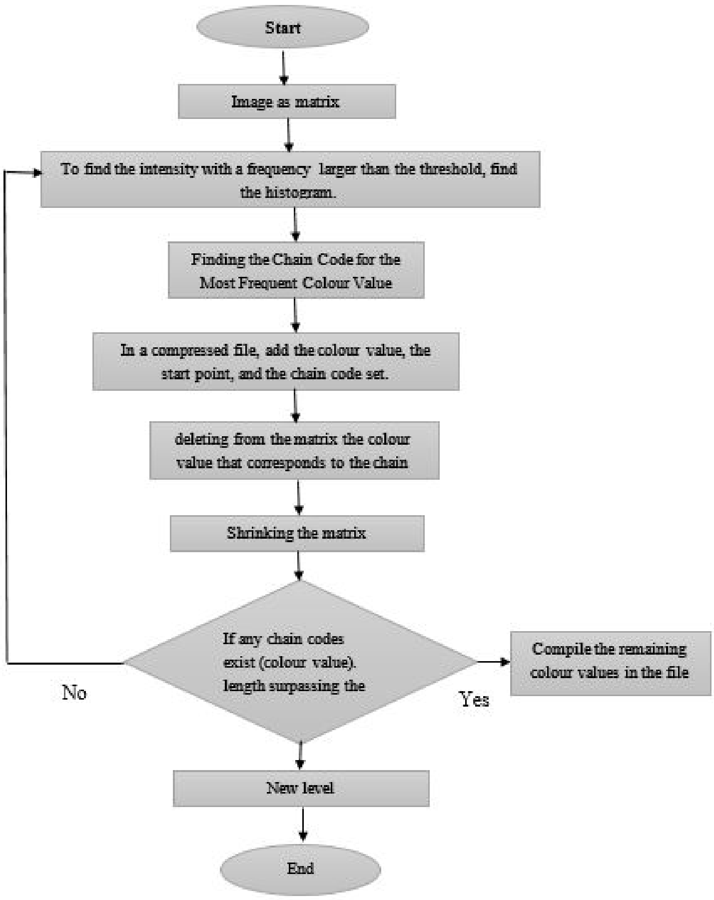

Figure 1. Overview of the proposed methodology

Use the clockwise 8-connected freeman chain code method to find the chain code for a particular colour value not accurate. Because that find one chain code value in every iteration from the neighbor. There are same colour value two or more in the neighbor, but it find the first value's chain code in the clockwise order so, it's take too much time and miss some values from the neighbor. That is, if colour values are arranged in the given format

When we use the clockwise chain code some particular colour values missed from neighbor and there is no drawback. That is a problem in this method. Use 8-connected freeman chain code method to find the chain code for a particular colour value accurate. Because that find one or more than one chain code value in every iteration from the neighbor. There are same colour value two or more in

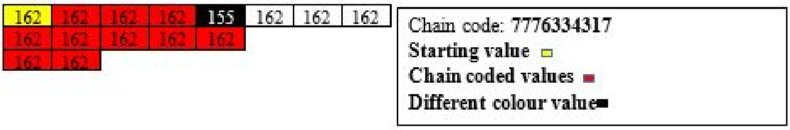

Figure 2

the neighbor, it find the all value in the neighbor so, it's take less time and no missing values from the neighbor for a particular colour value. That is, if colour values are arranged in the given format

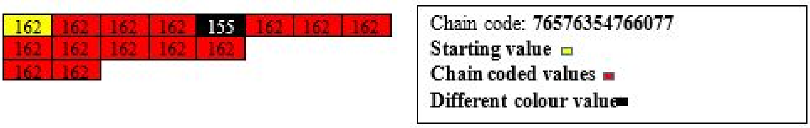

Figure 3

Problem with this method is difficult to do the reverse part to decompress the compressed file to image.

The proposed system consists of two parts, as follows:

- Compression
- Decompression

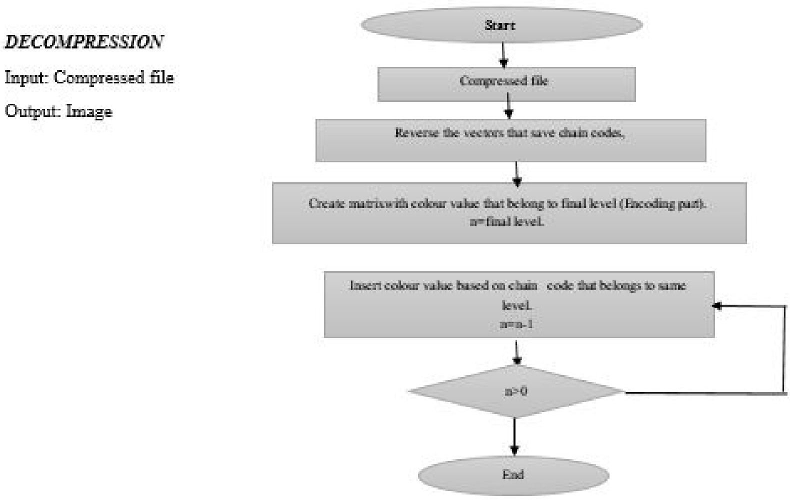

Figure 4. Flowchart of Decompression

The frequencies of colour values are discovered in the first section, followed by selection of the colour value with the highest frequency and finding a chain code for that colour value that meets the threshold condition. The colour value is then changed to a special value and the starting position of the colour value is changed in accordance with the chain code. The location of the starting point of the chain code, the colour value, and the chain code are then saved in a compressed file, after which we search for another chain code that belongs to the same colour.

To boost the likelihood of discovering new chain codes for colour values, we will eliminate special values and compress the input image by moving the colour values in the following step. These earlier steps are referred to as Level 1. Apply the same procedure as before to the image after it has been shrunk, and stop when there are no longer any chain codes in the shrunken image that meet the threshold length requirement.

I chose the threshold value of 5 for this system because it converts the 8-bit colour value to a 3-bit chain code. Therefore, if the threshold value is less than 3, it is pointless. In order to increase efficiency, it is desired to reduce the number of pixels as much as feasible; therefore, the threshold value should be more than 3 and lower than 8. As a result, I set the threshold value at 5. Technique for shrink image

- Move the values right to left
  In this method the image size is same as the input image size at every level. Because after move the remaining values right to left, remaining pixels are fill with border pixel value. Problem this method is algorithm spent more time with border valued pixel.

- Create a matrix and arrange the values in order
  In this method create an empty matrix with size square root of remaining values, and arrange the values in row by row order. Image size is change at each and every level. So I choose the second method to shrink the image in this system. The process is reversed in the second section compared to the compression process; the first step starts with values that are still in level n of compression, and the second step is based on start points, colour values, and chain codes; the image will grow when the colour values associated with a given level are returned to level 1, level 1, at which point we obtain the original image.

3 Results

Table 1. Compression results of images size 256*256=65536

Image	Number of Levels	Total Length of Chain code	Number of chain code
Jellyfish	6	39592	1126
Lighthouse	7	24029	1040
Hydrangeas	5	10945	692
Tulips	3	15700	529
Lenna	2	13052	930
Pepper	7	5053	591

The suggested approach is tested using experiments on 6 colour images with 256 colos (8 bpp), as seen in the figures below. The results of applying the suggested system to the 256*256 and 512*512 pixel images from above are shown in the table below.

Table 2. Compression results of images size 512*512= 262144

Image	Number of Levels	Total Length of Chain code	Number of chain code
Jellyfish	11	175402	3246
Lighthouse	22	121097	4184
Hydrangeas	5	79973	4619
Tulips	8	95227	4162
Lenna	7	23997	2665
Pepper	5	70430	6329

Table 3. Performance of individual classifiers after hyper parameter tuning and cross-validation with 5 folds

Levels	Starting Positons(X,Y)		Colour Value	Chain code
1	214	42	144	’5651456565’
1	222	40	144	’76534546556’
1	230	47	144	’561545654644’
1	212	45	141	’76536654565646565555555’
1	230	54	137	’62555555’
1	224	49	136	’763456’
1	236	50	140	’6207676546’
1	144	466	244	’765376767676767676767676767676’
2	142	131	15	’737776’
2	188	142	15	’465307’
2	192	48	15	’624533’
2	224	213	15	’465055’
2	226	170	15	’737777’
2	245	11	15	’737667’
2	274	202	15	’454052’
2	289	16	15	’470647’
2	37	173	17	’737777’
2	38	341	17	’4540432’
2	162	247	17	’737776’
2	174	211	17	’476540737’
2	194	215	17	’737777’
2	208	316	17	’430637’
3	43	116	216	’737776’
3	40	271	219	’4753073’
4	162	333	15	’736465’

4 Discussions and Conclusion

Based on the experimental findings, we remark the following: In this study, we presented a lossy image compression based on location by employing Freeman chain code representation.As image

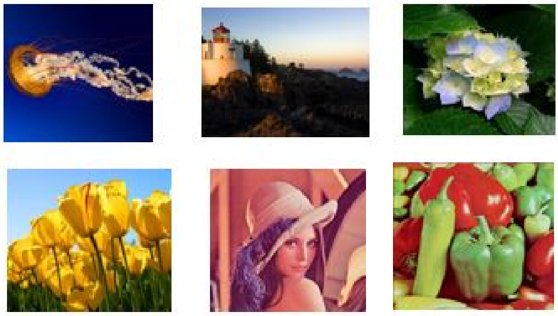

Figure 5

size increases, the compression ratio rises. The size of the image and the distribution of colour values within the image both affect how quickly an image can be compressed. The compression ratio is affected by threshold length in addition to chain code vector compression, and it also depends on how an image is shrunk. [1], [2], [3], [4], [5],[6],[7],[8],[9]

REFERENCES

[1] David Salomon. *Data compression: the complete reference*. Springer Science & Business Media, 2004.

[2] TC Bell, JG Cleary, and IH Witten. "Text Compression, Prentice Hall". In: *PTR: Englewood Cliffs, New Jersey* (1990).

[3] M Sindhu and R Rajkamal. "Images and its compression techniques-A Review". In: *International Journal of Recent Trends in Engineering* 2.4 (2009), p. 71.

[4] Rafael C Gonzalez, Richard Eugene Woods, and Steven L Eddins. *Digital image processing using MATLAB*. Pearson Education India, 2004.

[5] Sebastian Deorowicz. "Universal lossless data compression algorithms". In: *Philosophy Dissertation Thesis, Gliwice* (2003), p. 38.

[6] Chin-Chen Chang, Chih-Ping Hsieh, and Ju-Yuan Hsiao. "A new approach to lossless image compression". In: *International Conference on Communication Technology Proceedings, 2003. ICCT 2003.* Vol. 2. IEEE. 2003, pp. 1734–1738.

[7] Azzam Talal Sleit and Rahmeh Omar Jabay. "A Chain Code Approach for Recognizing Basic Shapes". In: *Information Technology College University Of Jordan, Amman, Jordan* (2006).

[8] Azzam Talal Sleit and Rahmeh Omar Jabay. "A Chain Code Approach for Recognizing Basic Shapes". In: *Information Technology College University Of Jordan, Amman, Jordan* (2006).

[9] Walid Shahab, Hazem Al-Otum, and Farouq Al-Ghoul. "A modified 2D chain code algorithm for object segmentation and contour tracing." In: *Int. Arab J. Inf. Technol.* 6.3 (2009), pp. 250–257.