

Image Compression with LZW (Lossless) On Different Image Formats

Prof.(Dr.) Hitesh H Vandra

Electronics and Communication, Shree Swaminarayan Institute of Technology, Bhat, Gujarat, India

ABSTRACT

Image compression is used to reduce bandwidth or storage requirement in image application. Mainly two types of image compression: lossy and lossless image compression. A Lossy Image Compression removes some of the source information content along with the redundancy. While the Lossless Image Compression technique the original source data is reconstructed from the compressed data by restoring the removed redundancy. The reconstructed data is an exact replica of the original source data. Many algorithms are present for lossless image compression like Huffman, rice coding, run length, LZW. LZW is referred to as a substitution or dictionary-based encoding algorithm. The algorithm builds a data dictionary of data occurring in an uncompressed data stream. Patterns of data (substrings) are identified in the data stream and are matched to entries in the dictionary. If the substring is not present in the dictionary, a code phrase is created based on the data content of the substring, and it is stored in the dictionary. The phrase is then written to the compressed output stream. In this paper we see the effect of LZW algorithm on the png, jpg, png, gif, bmp image formats.

Keywords : Image Compression, Lossless Image Compression, LZW, CoffeeScript

I. INTRODUCTION

Image compression serves the purpose of reducing bandwidth or storage requirement in image applications. There are two types of compression lossy and lossless. The Lossless Compression technique recommended preserves the source data accuracy by removing redundancy from the application source data. In the decompression processes the original source data is reconstructed from the compressed data by restoring the removed redundancy. The reconstructed data is an exact replica of the original source data. The quantity of redundancy removed from the source data is variable and is highly dependent on the source data statistics, which are often non-stationary. Lossless compression has the advantage of avoiding the issue whether the coding quality is sufficient So many coding are used

for lossless image compression like Huffman, run length, DPCM, LZW, etc. The original Lempel Ziv approach to data compression was first published in 1977. Terry Welch's refinements to the algorithm were published in 1984. The algorithm is surprisingly simple. In a nutshell, LZW compression replaces strings of characters with single codes. It does not do any analysis of the incoming text. Instead, it just adds every new string of characters it sees to a table of strings. Compression occurs when a single code is output instead of a string of characters. In this paper we see the effect of LZW algorithm on different image formats and compare the compression ratio for the same.

II. METHODS AND MATERIAL

2.1 LZW ALGORITHM LZW is referred to as a substitution or dictionary-based encoding algorithm.

The algorithm builds a data dictionary (also called a translation table or string table) of data occurring in an uncompressed data stream. Patterns of data (substrings) are identified in the data stream and are matched to entries in the dictionary. If the substring is not present in the dictionary, a code phrase is created based on the data content of the substring, and it is stored in the dictionary. The phrase is then written to the compressed output stream. When a reoccurrence of a substring is identified in the data, the phrase of the substring already stored in the dictionary is written to the output. Because the phrase value has a physical size that is smaller than the substring it represents, data compression is achieved. Decoding LZW data is the reverse of encoding. The decompressor reads a code from the encoded data stream and adds the code to the data dictionary if it is not already there. The code is then translated into the string it represents and is written to the uncompressed output stream.

LZW goes beyond most dictionary-based compressors in that it is not necessary to preserve the dictionary to decode the LZW data stream. This can save quite a bit of space when storing the LZW-encoded data. When compressing text files, LZW initializes the first 256 entries of the dictionary with the 8-bit ASCII character set (values 00h through FFh) as phrases. These phrases represent all possible single-byte values that may occur in the data stream, and all substrings are in turn built from these phrases. Because both LZW encoders and decoders begin with dictionaries initialized to these values, a decoder need not have the original dictionary and instead will build a duplicate dictionary as it decodes.

2.2 BLOCK DIAGRAM

Block diagram of the LZW compression is shown in the figure 1. It consist pre-processor, LZW encoder, LZW decoder, Postprocessor.

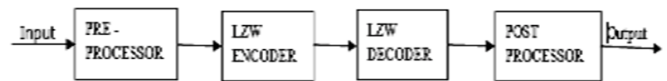


Figure 1 : Block Diagram of LZW lossless Image Compression

2.2.1 PREPROCESSOR

Preprocessor consists mapper and predictor. The mapper transforms the input data into a format designed to reduce intermixed redundancies in the input image. In this paper we used mapper which first take min value of the pixels and then subtract that value from each pixel's value. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image. The predictor is used to predict the next pixel value. Here we have used delta pulse code modulation as a predictor. The output of the predictor is given to the LZW encoder.

2.2.2 LZW ENCODER

Steps for the LZW encoding are shown below: -
 Initial table with initial character strings
 - P=first input character
 WHILE not end of input stream

C=next input character
 IF P+C is in the string table
 P=P+C
 ELSE
 Output the code for P
 Add P+C to the string table
 - P=C
 - END WHILE
 Output code for P

2.1.3 LZW DECODER

Steps for the LZW decoding are shown below:

- Initialize table with single character strings OLD = first input code

Output translation of OLD
 WHILE not end of input stream

- NEW = next input code
- IF NEW is not in the string table S = translation of OLD
- S = S+C
- ELSE
- S = translation of NEW
- Output S
- C = first character of S
- OLD + C to the string table

- OLD=NEW
- END WHILE

2.1.4 POSTPROCESSOR

Postprocessor consists inverse predictor and mapper. In the inverse predictor we have used DPCM decoder. Which first find the predicted value and get original value. In the inverse mapper we have done the reverse procedure of mapper. In inverse mapper. we subtract the min value which was in the mapper.

III. RESULTS

Table 1: Compression Ratio for different images

Main images	Different formats of main images				
	png	jpg	tif	gif	bmp
Lena	1.90	1.18	1.92	1.93	1.91
Gold-hill	1.08	1.09	1.08	4.15	1.06
Dog	2.20	2.23	2.23	5.50	2.23
Nature	1.61	1.60	1.61	4.40	1.60
Grey lone	1.51	1.33	1.51	3.82	1.51

IV. ORIGINAL IMAGES



Figure 2: Lena



Figure 3: Gold hills



Figure 4 : Dog



Figure 5: Nature



Figure 6: Grey lonesome

V. CONCLUSION

In the lossless image compression technique the reconstructed data is an exact replica of the original source data. Here we get zero error between original image and reconstructed image. From the results we can say that LZW provide higher compression ratio for the .gif format of any image than others. Also the LZW give the higher compression ratio for those images in which gray level changes gradually.

VI. REFERENCES

[1]. Gonzalez and Woods, "Digital Image Processing" 3rd edition, Prentice Hall, 2002, chapter-8.

- [2]. Mauro Barni, "Document and Image Compression", CRC, Taylor and Francis group, chapter-5.
- [3]. Khalid Sayood, "Lossless Compression Handbook", Academic Press, USA, chapter-6.
- [4]. Kou, Weidong. "Digital image compression : algorithms and standards", Boston, Kluwer Academic, 1995, chapter-2.
- [5]. Gonzalez and Woods, nd "Digital Image Processing using MATLAB" 2 edition, Prentice Hall, chapter-8.
- [6]. David Salomon, "Data Compression: The Complete Reference", 4th edition, Springer Verlag publication, page no. 195-206.
- [7]. Khaled S. Alkharabsheh, Prof. Ralph Oberly, Committee Chairperson, Ph.D, Prof. James Brumfield, Ph.D , "Image Compression And Its Effect On Data", Marshall University Aug. 02, 2004
- [8]. S.Sahni, B.C.Vermuri, C.Kapoor, "State of The Art Lossless Image Compression Algorithm"
- [9]. D.Wu and E.C.Tan, "Comparison of Lossless Image Compression Algorithm" School of Applied Science, Nan yang Technological University, Nanyang Avenue, Singapore 639798.
- [10]. CCITT, "Information Technology-Digital Compression And Coding Of Continuous Tone Still Image Requirements And Guidelines", ITU.

Cite this article as :

Dr. Hitesh H Vandra, "Image Compression with LZW (Lossless) On Different Image Formats ", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 3 Issue 3, pp. 2153-2156, March-April 2018. Available at doi : <https://doi.org/10.32628/CSEIT1952310>
Journal URL : <https://ijsrcseit.com/CSEIT1952310>