IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Comparative Analysis of Text and Image Compression Techniques Using LEMPEL-ZIV-WELCH Coding

Harshvardhan Singh
Gokula Prudhvi Kumar Yadav
Addagunta Srinath
Computer Science and Engineering
Lovely Professional University
Jalandhar, India

Abstract: The basic essence of Compression is to cut down the size and make the files to smaller size than they usually are. To get this process done right, data loss is the thing one has to look up to. Compression can be either lossy or loss-less. Having taken care of the data, memory consumption is a significant thing which always bothers & one should always look after this to yield productive results. The main objective & intent of this work is to check how Huffman Coding can result in making efficient use of memory while compressing text and image files, being lose-less.

I. INTRODUCTION

We live in a fast-paced world where the usage of messages and multimedia is at a higher extent. As if we take into consideration the social media, every user communicates with the other by texting and sending images. Files which we send must not be in larger size as it consumes a lot of bandwidth while transferring to the other end. Hence comes the necessity of compression, as it decreases not only the size but also makes an efficient usage of memory without any loss of data like-wise we have had prior to the compression. Quality of those files gets decreased but will eventually result in the retrieval of quality which may not be par-level but up to a certain extent which looks quite cool and same as to that of the original one. Compression is an approach that best suites whenever one finds it difficult to deal files that are quite big. If we look at compression techniques, the likes of Huffman Coding and Lempel-Ziv-Welch Coding algorithms are available to us with different approaches so as to increase the efficiency and also the performance speed.

II. TECHNIQUES AVAILABLE

Compression of Text :

Lempel-Ziv-Welch Coding: - It is a method which is good at compressing text files due to the reason it contains ASCII characters(stored as 8-bit binary values) whereas it doesn't hold good for graphics files as they have repeating patterns of binary digits.

To begin with, here we take an example with 6 character alphabets and a 16 entry dictionary. Let's say we have a message :

ababacdcdaaaaaaef

Given Dictionary is-

0000 'a'
0001 'b'
0010 'c'
0011 'd'
0100 'e'
0101 'f'
0110–1111 empty

0000 is sent with a character followed by b, sequence ab is checked by transmitter and ' sequence has been stored in the dictionary.

0000 'a'
0001 'b'
0010 'c'
0011 'd'
0100 'e'
0101 'f'
0110 'ab'
0111–1111 empty

The receiver adds to this table 'a' is read by transmitter and further checks 'ba'' sequence is in the code table. As it is not, it transmits the 'a' character as 0000, adds the 'ba' sequence to the dictionary, which will now contain:

0000 'a'
0001 'b'
0010 'c'
0011 'd'
0100 'e'
0101 'f'
0110 'ab'

0000 0001 0000 0110 0010
| | | | |
'a' 'b' 'a' 'ba' 'c'

0111 ‘ba’

1000–1111 empty

Character ‘b’ is ready by transmitter and checks presence of ‘ba’. Transmission of code table address i.e 0111 is done. Receiver checks the presence of the same in dictionary. Further transmitter reads a character c. It locates character c. It continues in iteration with the transmitter and receiver that maintains identical copies of their dictionaries.

Compression of Image :

Lempel-Ziv-Welch Coding and Run-Length Coding: - For understanding computer images, we need to know about bi-level(Two-level) images or binary images. A bi-level image is a raw computer image in which each pixel size is of one bit only, i.e, it can be either ON or OFF. Due to which, a bi-level image consists of two colours : a background colour and a foreground colour and in actual it is two states namely, Black and White.

In this run-length coding approach, we try to perceive the image firstly with all its pixels being distributed separately. Then after, we pick each pixel starting from the first and try to fill that pixel with either of the two colours black and white. To put any of the other colours, one has to follow the RGB band codes whose range is 0-255. Putting up the code of desired colour in the data packet(which has three fields to store that band codes). Usually for colour black, the code comes as (0,0,0) and it is (255,255,255) for colour white. This makes it to store colours in data packets. And we also need to check for any redundancy in pixels. Apart from that pixel, we need to move forward(iterating to each pixel) and check if there is any other pixel having same colour. When any repetition is encountered, redundancy needs to be reduced or eliminated. With run-length coding being done, we advance to compression.

During compression of an image, we use lempel-ziv-welch technique as it is quite simple and easy to code. After compressing the image, we send it to parsing, where a binary tree is generated to store the image in dictionary format i.e, key-value pair format. And that collection of key-value pairs is the output of our respective image. Hence this marks an end to image compression.

III. COMPARISON ANALYSIS

Different types of Compression techniques and Contrasting them:

Parameter	Lempel-Ziv-Welch Coding	Huffman Coding
Type	Dictionary-based	Entropy-code
No. of passes	One	Two
Complexity	Simple and Easy	Quite Complex
Execution	Faster	Slower than LZW
Prior info	Not required	Required
Versatility	Yes	No
Static Coding	No	Yes

PSEUDO CODE of Lempel-Ziv-Welch Coding

```
Initialize table with single character strings
    Ps = first input character
    while(!eos=end)
      Cs = next_input_character
    IF Ps + Cs is in the string table
        Ps = Ps + Cs
      ELSE
      output for Ps
       add Ps + Cs to the string table
      Ps = Cs
     END of WHILE
  Output code for Ps
```

IV. Conclusion and Future Work

Compressing aids us in reducing files' size by keeping almost same quality and making perfect use of memory. Indeed it's a most useful and significant technique which will probably need in the near future with some extra efficiency and opimizing tools. Depending on the day-to-day essence, this will be required an update. Meeting the expectations of users, this compression engine is made and for any changes or improvements to be included in this in future, the improvement could be improving the quality(no quality loss) of image of the output than we have right now and also decreasing the processing time with an increment in performance and speed. And lastly, we are about to include video compression which is very important.

In this modern world, shorter sized files are chosen over larger ones. Thus to have more free memory, everyone tries to use this compression and maintain the same quality as such in the upcoming years.

REFERENCES

[1] Ruey-Feng Chang,Wen-Tsuen Chen,and Jia-Shung Wang,"A Fast Finite-State Algorithm for Vector Quantization Design", IEEE Transaction on Signal Processing, Vol.40, No.1, January 1992.

[2] Hong Wong, Ling LU, DA-Shun Que, Xun Luo,"Image Compression Based on Wavelet Transform and Vector Quantization" IEEE proceedings of the First international Conference on Mchine Learning and Cybernetics, Beijing, November 2002.

[3] Y.W. Chen, Vector Quantization by principal component analysis, M.S. Thesis, National Tsing Hua University, June, 1998.

[4] M.F. Barnsley and L.P. Hurd, Fractal Image Compression, AK Peters, Ltd. Wellesley, Massachusetts, 1993.

[5] A.E. Jacquin, Imagecoding basedon a fractaltheory of iterated contractive image transformations. IEEE Trans. on Image Processing, vol. 1, 18-30, 1992.

[6] H.S. Chu, A very fast fractal compression algorithm, M.S. Thesis, National Tsing Hua University, June, 1997.

[7] Nitesh Kumar More,Sipi Dubey," JPEG Picture Compression Using Discrete Cosine Transform" International Journal of Science and Research (IJSR), India Online ISSN: 2319-7064

[8] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, Image coding using wavelet transform, IEEE Trans. on Image Processing, vol. 1, 205-220, 1992.

[9] A.S. Lewis and K. Knowles, Image compression using 2D wavelet transform, IEEE Trans. on Image Processing, vol. 1, 244-250, 1992.