Image Redundancy Compression using LZW Compression Algorithm

Siyabonga Magubane
UJ computer science honours student
Kingsway Rd and, University Rd, Auckland Park, Johannesburg, 2092
201589401@student.uj.ac.za

Abstract

Image compression algorithms include lossy and lossless compression. For image analysis and image processing applications there is a need for data transmission within applications that preserves image quality. An algorithm solution is proposed that uses LZW compression algorithm to compress and decompress images in real time. With image redundancy we can encode and decode images and without compromising image quality. We explore this by implementing a real time image compression and decompression application with the proposed algorithm using image spatial redundancy

1. Introduction

Various algorithms may be used for compressing images. There are namely two types of compression, loss compression where some data not visible to the human eye is neglected(lost) to compress the images reducing the size, such as JPEG compression algorithm. The second type of compression is the lossless compression where no data is sacrificed to compress the file. The file can be decompressed using various types of lossless algorithms

Transmission of data between application using data compression algorithms to decrease the amount of data transferred between the applications by enabling the sender to encode the data and the receiver to decode the data and interpret it. This concept gives room for us to develop a real time compression and decompression algorithm that uses redundancy within the image to compress the image. With this there is also a need for a high compression ratio algorithm that allows for the reproduction of the original image completely without losing any data.

2. Problem Background

The development of technological applications that can gain valuable information from images by extraction of data, analyzing it and gathering insight on the data from images have become more important in various scientific fields such as health, geographical information systems etc. The need to maintain 100% of the image quality during transmission of images among applications thus becomes more important.

Redundancy in images enables us to compress images with algorithms such as LZW compression, run-length, Huffman coding algorithms etc. There is among others three types of redundancy in images namely coding redundancy, spatial redundancy and irrelevant information. Coding redundancy refers to redundancy where only a gray value are present, and we can shorten the codewords by reducing the bits based redundant pattern within the image pixels. It is an inefficiency in data representation. An example of this is spatial redundancy and irrelevant information. Below is a model for image data compression and decompression.

Image compression and decompression model:

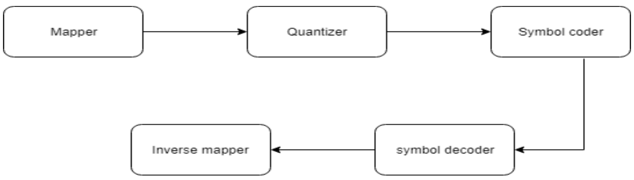


Figure 1. image compression model

Mapper transforms data to a form that can be compressed easier. This may be done by splitting pixels into matrices representing the raster image. Quantizer reduces amount of data where methods such as clustering to reduce psychovisual redundancies. The symbol coder simply codes the data using the shortest code words.

3. Solution Proposed

Applying the compression and decompression model using coding redundancy within image real time image compression and decompression may be achieved separating the image into image matrix that is split into 3 different matrixes representing pixels and applying LZW compression algorithm on these matrices. Below is a pseudo code that further explains this proposed algorithm.

1. ImageArray = ConvertImagetoArray(file_Path)
2. Matrix-R,Matrix-G,Matrix-B = ImageArray.split.()
3. Intialize Dictionary

4. Compress each matrix using redundancy algorithm
5. UpdateDictionary
6. Decompress each Matrix (Dictionary)
7. Combine Matrices
8. Convert to Image

This algorithm may be used to compress images in real-time and is lossless therefore the image will be reproduced once the compressed images are decompressed. Depending on the type of coding redundancy algorithm used the size of the dictionary may grow substantially. The compression ratio of this algorithm is dependent on the coding redundancy algorithm and the subject image. For example, in the experiment section below a spatial redundancy coding redundancy algorithm is used where a single row has the potential to represent the whole row of the matrix. Once the first element that represents the entire array matrix is found then it may be added onto the

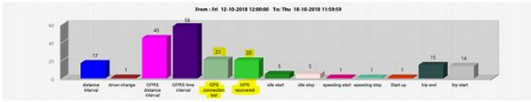
In this case if the image has various large areas of redundant color scheme the irrelevant information may be entirely represented by the first element in the row.

4. Experiment and results

The coding redundancy algorithm used is the spatial redundancy algorithm. Two different images bitmap images are used conduct test. Image 1 260X190px is a colorful image filled with different colors as per fig.2 below. Image 2 is 679X122px is a image with white area space.

Fig 2 Colorful Image

Fig 3 Image with large white area



The compression ratio with the below formula is used to calculate the efficiency of the compression.

$$Cr = Ni/Nc$$

Where Cr = Compression ration
Ni = number of bits before compression
Nc = number of bits after compression

Figure 1. compression ratio.

Table 1. Table caption.

Matrices	Image1	Image 2
r	1.0	1.101
g	1.0	1.102
b	1.0	1.101

5. Conclusion

Redundancy in images can be used to compress and decompress images without losing any data on the images enabling further analysis on images on different applications. Image compression efficiency using LZW compression efficiency is dependent on the redundancy algorithm used to compress the images and the image itself. The more the colors on the image the less effective is spatial redundancy code redundancy algorithm in compressing the subject image. The image height and width impact the average time taken to compress the image.

References

[1] Daniel Greene Mohan Vishwanath Frances Yao Tong Zhang *, 1997. A Progressive Ziv-Lempel Algorithm for Image Compression. Salerno, Italy, IEEE.

[2] Tony Lin , Pengwei Hao, 2005. Compound Image Compression for Real-Time. IEEE TRANSACTIONS ON IMAGE PROCESSING, 14(8), pp. 993-1005.

[3] University, U., 2005. Department of Computer Science. [Online]Available at: https://www8.cs.umu.se/kurser/TDBC30/VT05/material/lecture8.pdf[Accessed 30 10 2018]