Joanne Chen

Professor John C. Ramirez

CS1501

27th June 2018

In the table below, I compare four of the LZW variation programs for each of the test files. The four LZW variation programs include the LZW program that uses codewords of 12 bits (labeled as ‘LZW’ in the table), LZWmod program with variable length bits without dictionary reset (labeled as ‘LZW without reset’ in the table), LZWmod program with variable length bits and dictionary reset (labeled as ‘reset’ in the table) and compress.exe (labeled as ‘compress.exe’ in the table). The first column under “file size” lists the original file size in bytes of each test file. Before going further, I would like to explain data compression ratio. Data compression ratio is defined as the ratio between the original file size and compressed size. If compression ratio is greater than one, it means compressed data is usually smaller than uncompressed data, which is whole point to do compression. By comparing 14 test files using the four of the LZW variation programs, we could find out something useful. The 12-bits LZW program has compression ratio less than 1 for test files Lego-big.gif, frosty.jpg, winnt256.bmp and edit.exe. And for test files Lego-big.gif and frosty.jpg, LZW with variable length bits and LZW with variable length bits and dictionary reset have compression ratio less than 1. Even though compress.exe does not have compression ratio less than one for all the test files, it did almost zero compression for Lego-big.gif and frosty.jpg (has compression ratio almost 1).

Now we look closer to the table of which the test files gave the best and worst compression ratios. The original LZW program has worse compression ratio 0.71 when compressing frosty.jpg and best compression ratio 214.24 when compressing wacky.bmp. LZW with variable length bits has worse compression ratio by compressing Lego-big.gif and best compression ratio 233.27 by compressing wacky.bmp. And the worse compression ratio of LZW with variable length bits and dictionary reset is 0.71, given when compressing frosty.jpg. The best compression ratio of LZW with variable length bits and dictionary reset is 233.27 by compressing wacky.bmp. And frosty.jpg also gives worst compression ratio 1 and wacky.bmp gives best compression ratio 266.14 of the compress.exe.

Besides all the observations from above, I also find out that the following second and third best case compression ratio is always text file. And wacky.bmp is a simple monochrome picture, which contains many repetitive bytes. By contrast, frosty.jpg and Lego-big.gif are polychromatic. They contain less repetitive information and they grow bigger after compression. Thus, LZW compression often performs well with text and monochrome images, regardless if it has variable length bits or dictionary reset. This is because files that can build up very long patterns allow LZW to get a great deal of compression. And LZW compression is not a good option when compressing variant color images. LZW without dictionary reset would maintain long patterns that have been built up but would not do very well if file contains many changes. By contrast, if we reset dictionary, we would be able to compress new patterns, while compression is minimal until new patterns are added.

A picture containing indoor

Description generated with high confidence