UDC 004.921

IMAGE COMPRESSION METHOD BASED ON RUN LENGTH ENCODING

Wang Ying

gr.267011

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Tsviatkou Viktar, head of the department of infocommunication technologies

Annotation. Video compression is an important part of image manipulation. This article first briefly introduces several commonly used video compression methods. Then, based on the file format of BMP images, a compression program using run length encoding (RLE) and C language was proposed, and the compression effect was discussed. Finally, this article discusses the characteristics of RLE and proposes several reasonable methods to improve its compression performance.

Keywords: RLE, image manipulation, BMP

Introduction. Due to the large amount of data in images and videos, they bring a lot of inconvenience to storage and transmission. Therefore, image compression and video compression have been widely used. There are many types of data compression methods, which can be divided into lossless compression and lossy compression. Lossless compression uses statistical redundancy of data for compression, which can completely restore the original data without introducing any distortion. However, the compression rate is limited by the theoretical redundancy of data statistics, generally 2:1 to 5:1. The lossy compression method utilizes the insensitivity of human vision to certain frequency components in the image, allowing for the loss of certain information during the compression process; Although the original data cannot be fully restored, the lost part has a relatively small impact on understanding the original image, but has resulted in a much larger compression ratio [1]. It should be pointed out that the compression methods involved in this article are all lossless compression (with recoverability). There are several commonly used lossless compression methods for images:

RLE: Run length encoding is one of the simplest methods for compressing a file. Its approach is to replace a series of repeated values (such as the grayscale values of image pixels) with a single value and a count value. This method is easy to implement and effective for compressing strings with long repeated values. For example, for images with large areas of continuous shadows or the same color, The compression effect is very good using this method.

Arithmetic encoding: Arithmetic encoding is one of the main algorithms for image compression. This is a lossless data compression method and also an entropy encoding method. Unlike other entropy coding methods, other entropy coding methods typically divide the input message into symbols and then encode each symbol, while arithmetic coding directly encodes the entire input message into a number, a decimal n that satisfies ($0.0 \leqslant n<1.0$).

LZW encoding: The principle is to pair the value of each byte with the value of the next byte as a character pair, and set a code for each character pair. When the same character pair reappears, it is replaced by a code, and then paired with the next character. Its code can not only replace a string of data with the same value, but also, It can also replace a string of data with different values. If there are some data with different values that frequently appear repeatedly in image data, a code can be found to replace these data strings[2].

Huffman coding: Huffman encoding is achieved by replacing the original data with an unfixed length encoding. It was originally established to compress text files, but now there are many variants. Its basic idea is that values with higher frequencies correspond to shorter encoding lengths, while values with lower frequencies correspond to longer encoding lengths.

BMP is the abbreviation of Bitmap, which supports RGB color mode, index mode, grayscale, and bitmap color mode. It is the standard image file format in the Windows operating system and can be supported by various Windows applications. With the popularity of the Windows operating system and the development of rich Windows applications, BMP bitmap format is naturally widely used. The characteristic of this format is that it contains rich image information, which is almost not compressed, but this has resulted in its inherent disadvantage of occupying too much disk space[3]. In a BMP image file, the first 54 bytes of space are used to write basic information about the file, which can be divided into two parts: the file header (14 bytes) and the file information header (40 bytes).

In addition to the basic information of the file header and file information header, the content of the image body part of the BMP file is recorded line by line in the form of dots. The color information of each dot is recorded in three bytes, which respectively record the saturation of B (blue), G (green), and R (red) colors saturation, expressed as integers ranging from 0 to 255. In addition, the number of bytes recorded for each line of point element information in the BMP file must be a multiple of four. If the actual point element information is not a multiple of

four, a few bytes of zeros must be added at the end of each line.

Implementation of compressed programs. The C source code uses run length encoding to compress BMP images. The principle is to first read a point as a reference after processing the basic information of the file, and then read the next point to see if it has the same value as the reference point. If it is the same, add the statistical number of the reference point by one, and continue to count the number of points with the same value. When a new point with a different value from the reference point is read, the new point is used as the reference point, and the pixels in the image are read in this way until the end of the file.

Explore and improve methods. The compression ratio that RLE can achieve mainly depends on the characteristics of the image itself. If the image has larger blocks of the same color and fewer blocks, the compression ratio obtained will be higher. Conversely, RLE may not be able to handle natural images with rich colors, and there are often very few continuous pixels with the same color on the same line, And the number of consecutive rows with the same color value is even fewer. If RLE encoding method is still used, not only can image data not be compressed, but it may also make the original image data larger. Therefore, in specific implementation, it needs to be applied in conjunction with other compression encoding techniques.

Reading Point Path: The C program example given above reads points in line (main) order. Using this encoding method to compress bitmaps with no or minor changes in the horizontal direction can achieve a higher compression ratio. However, in practical situations, we often encounter images with more changes in the horizontal direction but the same pixel values in the local block range. In this case, We can use two other encoding methods for reading point paths, as shown in Figures 1 and 2:

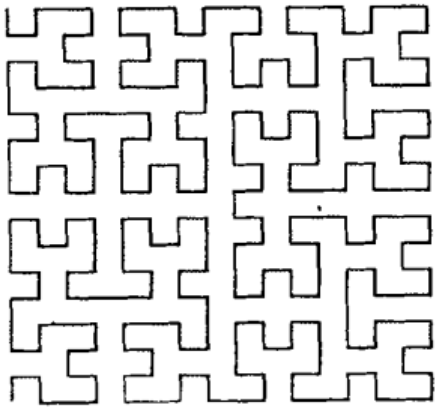

Figure 1 – Morton sequence

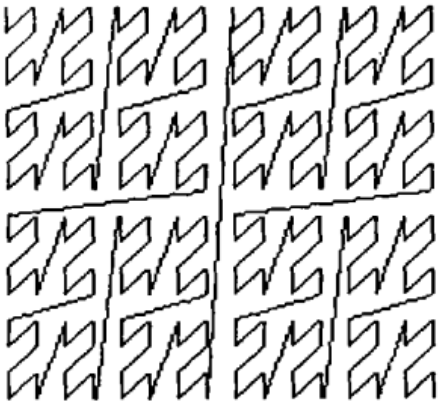

Figure 2 – Pi sequence

The use of the above two types of read point path encoding can achieve good compression ratios when compressing block like images with the same value. The phenomenon of block like images with the same value is common in various images, but the complexity and length of the program will increase significantly when using the above two read point paths.

Establishing a color index: In the above program, we can see that in the compressed file, a set of identical points should be recorded for at least four bytes (three bytes are used to record the saturation of R (red), B (blue), and G (green) colors, that is, the color values of the same value points in that group, and the other byte is used to record the number of same value points), If one or several sets of points of the same value with different sizes appear frequently

in the image, a considerable amount of space in the compressed file is used to repeatedly record the color values of these groups of points. To avoid wasting this resource, we can establish a color index, which is a shorter encoding to replace a color value. For example, if we want to record 40 positive red points, we originally need to write 0000FF 28 four bytes, When we use 1 to do the color index code for positive red, only 0128 bytes are needed. This means that when using run through encoding to compress an image with few color values, we can first traverse each pixel value of the image and index each different pixel value with a shorter code[4]. This way, when recording a set of same value points in the compressed file, Only using the index of the same value point color values can improve the compression ratio to a certain extent. It is not difficult to see why using run length encoding to compress binary images achieves better results.

Conclusion. Firstly, the image can be compressed by changing the order of the pixels in the read image, such as Morton sequence and Pi sequence. These two types of read path encoding can achieve better compression ratios when compressing block like images with the same value. Secondly, establish a color index. When compressing an image with low color values, using a shorter code as the index can also improve the compression ratio.

References

[1] Rong Guan'ao. Computer Image Processing [M]. Beijing: Tsinghua University Press, 2000.
[2] Statement Peng. Introduction to Geographic Information Systems [M]. Beijing: Science Press, 1997.
[3] Tan Haoqiang. C Language Programming [M]. Beijing: Tsinghua University Press, 2001.
[4] Wayne E. Carlson. A Survey of computer Graphics Image Encoding and Storage Formats [J]. Computer Graphics, 1991, 25(2) :20-23.

UDC 004.921

МЕТОД СЖАТИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ КОДИРОВАНИЯ ПУТЕШЕСТВИЙ

Ван Ин

гр.267011

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Цветков Виктор Юрьевич, заведующий кафедрой, профессор кафедры ФИБ

Примечания. Сжатие видео является важной частью обработки изображений. Эта статья начинается с краткого описания нескольких распространенных методов сжатия видео. Затем формат файла, основанный на изображениях BMP, предлагает программу сжатия, которая использует кодирование путешествий (RLE) и язык C, и обсуждает эффект сжатия. Наконец, в этой статье обсуждаются характеристики RLE и предлагаются несколько разумных способов улучшить его производительность сжатия.

Ключевые слова: RLE, Обработка изображений, BMP