www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issues 8 Aug 2016, Page No. 17535-17539

Image Compression - An Overview

Jagroop Singh[1]
[1]Faculty DAV Institute of Engineering and Technology, Kabir Nagar, jalandhar.

roopsidhu@yahoo.com

***Abstract:** With the continuing growth of modern communications technology, demand for digital image transmission and storage is increasing rapidly. Digital technology has given a great benefit to the area of image processing. Image compression is an economically viable alternative to these constraints by reducing the bit size required to store and represent the images, while maintaining relevant information content. The improvement of computer hardware including processing power and storage power has made it possible to utilize many-sophisticated signal processing techniques in advanced image compression algorithms.*

Keywords: *Bit rate, coding, Redundancy, deblocking.*

1 Introduction

With the continuing growth of modern communications technology, demand for digital image transmission and storage is increasing rapidly. Digital technology has given a great benefit to the area of image processing. Image compression is an economically viable alternative to these constraints by reducing the bit size required to store and represent the images, while maintaining relevant information content. The improvement of computer hardware including processing power and storage power has made it possible to utilize many-sophisticated signal processing techniques in advanced image compression algorithms.

This paper covers the background information on image compression.

2 Image Compression Framework

Image processing is very difficult because of large amounts of data used to represent an image. Technology permits ever increasing image resolution (spatially and gray levels), and increasing number of spectral bands. Consequently, there is an urgent need to limit the resulting data volume. One possible approach to decrease the necessary amount of storage is to work with compressed image data. Generally, image data compression is concerned with minimization of the number of information carrying units used to represent an image Jain (1981) [1]. Image compression takes advantage of the fact that there is a lot of redundant information contained in the original image. The purpose of image compression is to remove redundancies that take up valuable storage space and transmission time. These data redundancies have no contribution to the quality of the image and carry no additional information. In general, they can be classified as inter-pixel, coding, psycho-visual as shown in Table 1.

- Inter-pixel Redundancy: The images are not just random pixels; they have many kinds of structures i.e., there are statistical dependencies between pixels, especially between neighboring pixels. This kind of dependence is a redundancy that may be removed to achieve compression.

Table 1: Digital data redundancies

Redundancies	Descriptions
Inter-pixel	Information redundancy between pixels and pixels within the same image frame
Coding	Information redundancy within the series of code that represent the image.
Psycho visual	Information that represents detail which is not perceivable by the HVS.

Fundamentally, there are two types of image data compression: lossless or reversible compression, and lossy or irreversible compression. While in the former, very little information is lost when the image is decompressed or reconstructed, and there is some loss of information in the latter.

While lossless compression offers limited compression ratios generally between 2:1 to 4:1, irreversible (lossy) compression algorithms can provide higher compression ratios as high as 10:1 or higher. It is felt that irreversible compression would serve a more useful purpose in terms of

DOI: 10.18535/ijecs/v5i8.18

reducing storage space and decreasing image transmission times, but maintaining image quality is also of primary importance.

3 Lossless Compression

Lossless compression techniques involve no loss of information. Lossless compression is generally used for applications that cannot tolerate any difference between original and reconstructed data. Examples of lossless methods are Run Length coding, Huffman coding, Lempel/Ziv algorithms, and Arithmetic coding.

3.1 Run Length Encoding

Run length encoding, sometimes called recurrence coding, is one of the simplest data compression algorithms. It is effective for data sets that are comprised of long sequences of a single repeated character. For instance, text files with large runs of spaces or tabs may compress well with this algorithm as suggested by Sayood (2000) [2].

3.2 Huffman Coding

Huffman coding, developed by Huffman (1952) [3], is a classical data compression technique. It has been used in various compression applications, including image compression. It uses the statistical property of characters in the source stream and then produces respective codes for these characters. These codes are of variable code length using an integral number of bits. The codes for characters having a higher frequency of occurrence' are shorter than those codes for characters having lower frequency. This simple idea causes a reduction in the average code length, and thus the overall size of compressed data is smaller than the original. Huffman coding is based on building a binary tree that holds all characters in the source at its leaf nodes, and with their corresponding character's probabilities at the side.

3.3 Lempel-Ziv-Welch (LZW) Encoding

This original approach is given by Ziv and Lempel (1977) [4]. Refinements to the above algorithm were published in 1984 by Welch (1984) [5]. 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.

The code that the LZW algorithm outputs can be of any arbitrary length, but it must have more bits in it than a single character. The first 256 codes (when using eight bit characters) are by default assigned to the standard character set. The remaining codes are assigned to strings as the algorithm proceeds.

3.4 Arithmetic Coding

Arithmetic coding is also a kind of statistical coding algorithm similar to Huffman coding. However, it uses a different approach to utilize symbol probabilities, and performs better than Huffman coding. In Huffman coding, optimal codeword length is obtained when the symbol probabilities are of the form $(1/2)^x$, where x is an integer. This is because Huffman coding assigns code with an integral number of bits. This form of symbol probabilities is rare in practice. Arithmetic coding is a statistical coding method that solves this problem. The code form is not restricted to an integral number of bits. It can assign a code as a fraction of a bit.

Therefore, when the symbol probabilities are more arbitrary, arithmetic coding has a better compression ratio than Huffman coding. In arithmetic coding, a one-to-one correspondence between the source symbols and code words does not exist. Instead, entire message is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between 0 and 1. As the message becomes longer, the interval needed to represent it becomes smaller, and the number of bits needed to specify that interval grows. Successive symbols of the message reduce the size of the interval in accordance with the symbol probabilities generated by the model. The more likely symbols reduce the range by less than the unlikely symbols and hence add fewer bits to the message. Before anything is transmitted, the range for the message is the entire interval [0, 1], denoting the half-open interval $0 \le x < 1$. As each symbol is processed, the range is narrowed to that portion of it allocated to the symbol.

In an arithmetic coder, the exact symbol probabilities are preserved, and so compression effectiveness is better, sometimes distinctly so. On the other hand, use of exact probabilities means that it is not possible to maintain a discrete code word for each symbol; instead an overall code for the whole message must be calculated.

For additional information on arithmetic coding the reader is referred to Witten et al. (1987) [6], Moffat et al. (1998) [7].

4. Lossy Compression

Lossy compression techniques involve some loss of information, and data that have been compressed using lossy techniques generally cannot be recovered or reconstructed exactly. In return for accepting this distortion in the reconstruction, much higher compression ratios can generally be obtained than is possible with lossless compression. In many applications, this lack of exact reconstruction is not a problem.

In brief, it can be said that unlike lossless compression algorithms, the lossy compression algorithms are based on concept of compromising the accuracy of the data in exchange for increased compression. If the resulting distortion (which may or may not be visually apparent) can be tolerated, the increase in compression can be significant. In the present work, mainly, the lossy techniques have been dealt in the later sections.

4.1 Vector Quantization

Vector Quantization (VQ) uses a codebook containing pixel patterns with corresponding indexes on each of them. The main idea of VQ is to represent arrays of pixels by an index in the codebook. In this way, compression is achieved because the size of the index is usually a small fraction of that of the block of pixels.

The main advantages of VQ are the simplicity of its idea and the possible efficient implementation of the decoder. Moreover, VQ is theoretically an efficient method for image compression, and superior performance will be gained for large vectors. However, in order to use large vectors, VQ becomes complex and requires many computational resources (e.g. memory, computations per pixel) in order to efficiently construct and search a codebook. More research on reducing this complexity has to be done in order to make VQ a practical image compression method with superior quality as proposed by Sayood (2000).

4.2 Predictive Coding

Predictive coding has been used extensively in image compression. Predictive image coding algorithms are used primarily to exploit the correlation between adjacent pixels. They predict the value of a given pixel based on the values of the surrounding pixels. Due to the correlation property among adjacent pixels in image, the use of a predictor can reduce the amount of information bits to represent image.

This type of lossy image compression technique is not as competitive as transform coding techniques used in modern lossy image compression, because predictive techniques have inferior compression ratios and worse reconstructed image quality than those of transform coding.

4.3 Fractal Compression

The application of fractals in image compression was first given by Barnsley et al. (1988) [8]. Fractal image compression is a process to find a small set of mathematical equations that can describe the image. By sending the parameters of these equations to the decoder, we can reconstruct the original image.

In general, the theory of fractal compression is based on the contraction mapping theorem in the mathematics of metric spaces. The Partitioned Iterated Function System (PIFS) which is essentially a set of contraction mappings, is formed by analyzing the image. Those mappings can exploit the redundancy, which is commonly present in most images. This redundancy is related to the similarity of an image with itself i.e., part A of a certain image is similar to another part B of the image, by doing an arbitrary number of contractive transformations that can bring A and B together. These contractive transformations are actually common geometrical operations such as rotation, scaling, skewing and shifting. By applying the resulting PIFS on an initially blank image iteratively, we can completely regenerate the original image at the decoder. Since the PIFS often consists of a small number of parameters, a huge compression ratio (500 to 1000 times) can be achieved by representing the original image using these parameters. However, fractal image compression has its own disadvantages. Because fractal image compression usually involves a large amount of matching and geometric operations, it is time consuming. The coding process is so asymmetrical that encoding of an image takes much longer time than decoding.

5. Hybrid Compression

Hybrid compression refers to an encoding method that combines the lossy and lossless compression in one encoding block. The advantage of hybrid compression comes from combining the advantages of lossless and lossy compression, at the same time reducing the limitations of lossless and lossy compression. The compression rate for lossy compression is an irreversible process and non-essential details of the image will be truncated. Since the process includes quantization, the reconstructed image deviates from the original image. In terms of removal of redundancy, lossy compression encoding technique targets mainly on reducing spatial, coding, and psychovisual redundancies. Nonetheless, as a compensation for the loss in image quality, lossy compression gives a much higher compression ratio. Lossy compression for still hybrid compression lies between lossless and lossy compressing rates and it is strongly dependant of the image characteristics (type, amount of color and details) as well as the encoding methods used for the containing blocks.

6. Joint Photographic Experts Group (JPEG)

JPEG is a very popular continuous tone (monochrome and color), still-frame compression standard. Its popularity is mainly due to its capability in maintaining a significant compression rate at an acceptable image quality, in comparing with many lossless compression techniques. The JPEG algorithm focuses on the removal of nonessential details, which are psychovisually not perceivable to the HVS. An image transform technique that de-correlates pixel intensities in this small spatial region (e.g. 8×8 blocks) and packs the energy in as few coefficients as possible would be ideally suited for this task, because in this case, a good approximation to the original data would be possible by coding only the few high energy coefficients. The Karhunen-Loeve Transform (KLT) is the statistically

DOI: 10.18535/ijecs/v5i8.18

optimal transform for this task. However, it is signal dependent and is computationally very involved. Instead, the DCT is widely used in practice. For most natural images, the DCT is also shown to be very effective at de-correlation and energy compaction. In particular, the DCT results in high energetic low spatial frequency components for most natural images since they possess significant low frequency content. Furthermore, fast algorithms for the computation of the DCT are available.

Fig. 2 shows the block diagram of the JPEG compression procedures. The JPEG compression algorithm involves a *compressor* and an *encoder*. The compressor consists of three sequential steps: -2^{n-1} level shifting, discrete cosine transformation and quantization. After compression, the data will be encoded by variable length coding. During compression, the image is first divided into 8×8 subimage blocks. Then, each of these subimage blocks will undergo the compression algorithm with level shifted, transformed and quantized individually. For each block, quantization of the DCT coefficients is performed using a quantization table specifying possibly different quantization intervals for each of the 64 DCT coefficients.

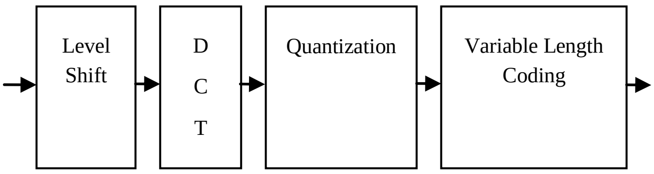

Fig. 2 Block diagram of JPEG compression system

Many of the 64 DCT coefficients in a block may be quantized to zero. Furthermore, for most blocks, the nonzero quantized coefficients will reside in the lower spatial frequency region. Common practice is to scan one block of quantized DCT coefficients in zigzag fashion starting from the lowest spatial frequency coefficient to the highest spatial frequency coefficient by Gonzales and Woods (2003) [9]. The resulting string of quantized coefficients undergoes run-length encoding. Run-length encoding produces symbols that carry two pieces of information: number of zeros in the run, and the nonzero value that terminates the run. These symbols are coded using variable-length-codes (VLC). Huffman codes are the most frequently used VLCs. Arithmetic codes are another choice. To reconstruct the image at the decoder, reverse of these steps must be performed.

7. Motion Picture Expert Group (MPEG)

The MPEG is a series of video coding standards developed by motion picture expert group – a collection of researchers from academia, industry, and other research organizations. Video and associated audio data can be compressed using MPEG compression algorithms. Three standards are frequently cited:

- MPEG-1 for compression of low-resolution (320×240) full-motion video at rates of 1-1.5 Mb/s by Le Gall (1992).
- MPEG-2 for higher resolution standards such as TV and HDTV at rates of 2-80 Mb/s by ISO/IEC 13818-2 (1993).
- MPEG-4 for small-frame full motion compression with slow refresh needs, at rates of 9-40 kb/s for video telephony and interactive multimedia such as video conferencing by MPEG video group (1997).

In video, both spatial and temporal redundancies are present. Exploiting only spatial redundancies (for example by applying JPEG to each frame) is known as intra-coding. Exploiting temporal redundancies between frames is called inter-frame coding. Using inter frame compression, compression ratios of 200 can be achieved in full-motion, motion-intensive video applications maintaining reasonable quality. A frame structure is defined, known as a group of pictures (GOP), which consists of a base intra-coded frame (I), a number of forward predicted inter-coded frames (P), and several bi-directionally predicted inter-coded frames (B) predicted from adjacent I and P frames. For predicted frames, motion within a spatial area (usually 16×16) known as macro-block is determined through motion estimation (local correlation). The motion vectors are encoded, and the prediction subtracted from the input frame, with only the difference transmitted using a JPEG type coding. Many of these differences are small or zero, enabling significant data reduction as suggested by Pereira and Ebrahimi (2002) [10].

8. Conclusion

Data compression techniques are required because of two main reasons: effective storage and efficient real time transmission of information. Lossless algorithms are used for the applications that require an exact reconstruction of original data while lossy compression is used when the user can tolerate some distortion. Compression ratio of lossless algorithms is significantly less than that of lossy algorithms.

References

[1] *A. K Jain, "Image data compression: a review," Proceedings of IEEE, vol. 69, pp. 349-389, 1981.*

[2] *K. Sayood, "Introduction to data compression," 2nd edition, Harcourt India Private Limited, 2000.*

[3] *D. A. Huffman, "A method for construction of minimum-redundancy codes," Proceedings IRE, vol. 40, pp. 1098-1101, 1952.*

[4] *J. Ziv, A. Lempel, "A universal algorithm for sequential data compression," IEEE Transactions on Information Theory, vol. IT-23, no. 3, pp. 337-343, 1977.*

[5] *T. A. Welch, A technique for high performance data compression," IEEE Transactions on Computer, vol.17, no. 6, pp. 8-19, 1984.*

[6] *I, H. Witten, R. M. Neal, J. G. Cleary, "Arithmetic coding for data compression," Communication ACM, vol. 30, no. 6, pp. 520-540, 1987.*

[7] *A. Moffat, R. M. Neal, I. H. Witten, "Arithmetic coding revisited," ACM Transactions on Information Systems," vol. 16, no. 3, pp. 256-294, 1998.*

[8] *M. F. Barnsley, A. Jacquin, Malassenet, F., Reuter, L., Sloan, A. D., "Harnessing chaos for image synthesis," Computer Graphics, vol. 22, no. 4, pp. 131-140, 1988.*

[9] *R. C. Gonzales, and R. E. Woods, Digital Image Processing. 3rd edition, New Delhi, Pearson Education, 2003.*

[10] *F .C. N. Pereira, T. Ebrahimi, The MPEG-4 Book, New Jersey, IMSC Prentice Hall Professional, 2002.*