A brief Introduction to Lossless Image Compression And Digital Watermarking

Muhammad Zubair Hasan and Shahir Abdullah

Department of Computer Science Engineering
Military Institute Of Science And Technology
Mirpur Cantonment, Dhaka-1216

mdzh1997@gmail.com

***Abstract* - Digital image compression has been the focus of a large amount of research in recent years. As a result, Image compression methods grow as new algorithms or variations of the already existing ones are introduced. In order to utilize digital images effectively, specific methods are needed to reduce the number of bits required for their presentation. It has led to an rapid growth in Area of Digital Image Processing. In image compression, we do not only concentrate on reducing size but also concentrate on doing it without losing quality and information of image. In our project we only implemented various lossless image compression techniques.**

On the other hand a digital watermark is a kind of marker covertly embedded in a noise-tolerant signal such as an audio, video or image data. It is typically used to identify ownership of the copyright of such signal. "Watermarking" is the process of hiding digital information in a carrier signal; the hidden information should,[1] but does not need to, contain a relation to the carrier signal. Digital watermarks may be used to verify the authenticity or integrity of the carrier signal or to show the identity of its owners. It is prominently used for tracing copyright infringements and for banknote authentication.

Index Terms -

1. Introduction

An image is essentially a 2-D signal processed by the human visual system. The signals representing images are usually in analog form. However, for processing, storage and transmission by computer applications, they are converted from analog to digital form. A digital image is basically a 2-Dimensional array of pixels. Image compression is a method through which we can reduce the storage space of images, videos which will helpful to increase storage and transmission process's performance.Image compression may be lossy or lossless. Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. But in lossy compression techniques, some of the finer details in the image can be sacrificed for the sake of saving a little more storage space.

Like traditional physical watermarks, digital watermarks are often only perceptible under certain conditions, i.e. after using some algorithm.[2] If a digital watermark distorts the carrier signal in a way that it becomes easily perceivable, it may be considered less effective depending on its purpose.[2] Traditional watermarks may be applied to visible media (like images or video), whereas in digital watermarking, the signal may be audio, pictures, video, texts or 3D models. A signal may carry several different watermarks at the same time. Unlike metadata that is added to the carrier signal, a digital watermark does not change the size of the carrier signal.

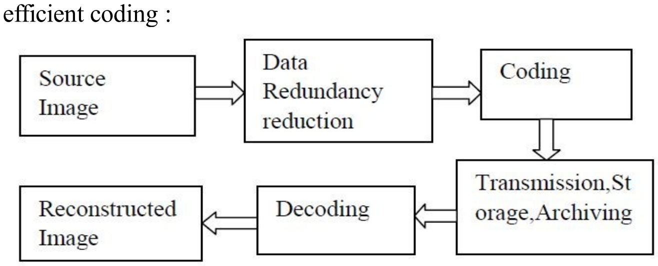
The needed properties of a digital watermark depend on the use case in which it is applied. For marking media files with copyright information, a digital watermark has to be rather robust against modifications that can be applied to the carrier signal. Instead, if integrity has to be ensured, a fragile watermark would be applied.

Both steganography and digital watermarking employ steganographic techniques to embed data covertly in noisy signals. While steganography aims for imperceptibility to human senses, digital watermarking tries to control the robustness as top priority.

Since a digital copy of data is the same as the original, digital watermarking is a passive protection tool. It just marks data, but does not degrade it or control access to the data.

2. Image Compression

The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form.Compression is achieved by the removal of one or more of three basic data redundancies
(1)Coding redundancy, which is present when less than optimal (i.e. the smallest length) code words are used.
(2) Interpixel redundancy, which results from correlations between the pixels of an image
(3) Psycho visual redundancy, which is due to data that is ignored by the human visual system (i.e. visually non essential information).
Image data Compression exploits redundancy for more efficient coding :



2.1 HOW MOST IMAGE COMPRESSION TECHNIQUES WORK

The usual steps involved in compressing an image are

1. Specifying the Rate (bits available) and Distortion (tolerable error) parameters for the target image.
2. Dividing the image data into various classes, based on their importance.
3. Dividing the available bit budget among these classes, such that the distortion is a minimum.
4. Quantize each class separately using the bit allocation information derived in step3.
5. Encode each class separately using an entropy coder and write to the file. Reconstructing the image from the compressed data is usually a faster process than compression. The steps involved are
6. Read in the quantized data from the file, using an entropy decoder. (Reverse of step 5).
7. Dequantize the data. (Reverse of step 4).
8. Rebuild the image. (Reverse of step 2).

2.2 Why do we need compression?

- Sufficient amount of storage space increased.
- To reduce the time taken for transmission of an image to be sent over the internet or download from the web pages.
- Image Archiving :Satellite Data
- Image Transmission: Web Data
- Multimedia Applications: Desktop Editing

3. Lossless Image Compression Techniques

Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art etc. Methods for lossless image compression are:

- Run-length encoding – used as default method in PCX and as one of possible in BMP, TGA, TIFF
- DPCM and Predictive Coding
- Entropy encoding
- Adaptive dictionary algorithms such as LZW – used in GIF and TIFF
- Deflation – used in PNG, MNG, and TIFF
- Chain codes
- Huffman Encoding

In our project we used following **compression techniques:**

i) Run Length Encoding

RLE is a lossless data compression technique which replaces data by a(length, value)pair, where value is the repeated value and length is the number of repetitions. RLE is a simple form of data compression in which data is in the form of runs. Runs is sequences in which the same data value occurs in many consecutive data elements are stored as a single data value and count, rather than as the original run. For example, consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text.

Consider a single scan line, with B representing a black pixel and W white pixel :

WWWWWWWWWWWWBWWWWWWWWWWW
WWWBBBWWWWWWWWWWWWWWWWWW
WWWWWWWBWWWWWWWWWWWWWW

If we apply the run-length encoding (RLE) data compression algorithm to the above hypothetical scan line, we get the following:

12W1B12W3B24W1B14W

This is to be interpreted as twelve Ws, one B, twelve Ws, three Bs, etc.

The run-length code represents the original 67 characters in only 18. **.** But, the actual format used for the storage of images is generally binary rather than ASCII characters like this, but the principle remains the same.

ii) Predictive Coding

Lossless predictive coding predicts the value of each pixel by using the value of its neighboring pixels. Therefore, every pixel encoded with a prediction error rather than its original value. These errors are much smaller compared with original value so that fewer bits are required to store them. Such as DPCM (Differential Pulse Code Modulation) is a lossless coding method, which means that the decoded image and the original image have the same value for every corresponding element. A variation of the lossless predictive coding is adaptive prediction that splits the image into blocks and computes the prediction coefficients independently for each block to high prediction performance.

iii) Huffman coding

Huffman Coding is an entropy encoding algorithm used for lossless data compression. Huffman Coding utilizes a variable length code in which short code words are assigned to more common values or symbols in the data, and longer code words are assigned to less frequently occurring values. The Huffman's algorithm is generating minimum redundancy codes as compared to other algorithms. The Huffman coding has effectively used in text, image, video compression, and conferencing system such as, JPEG, MPEG-2, MPEG-4, and H.263 etc..

iv) Discrete Cosine Transform (DCT)

DCT is a lossy Compression technique which is widely used in area of image and audio compression. Example: JPEG Images.DCTs are used to convert data into the summation of series of cosine waves oscillating at different frequencies. These are very similar to Fourier Transforms, but DCT involes use of Cosine functions and real coefficients, Fourier Transforms use both sine and cosine functions and complex numbers. For compression, Cosine functions are much more efficient as fewer functions are needed to approximate a signal. Both Fourier and DCT convert data from a spatial domain into a frequency domain and their respective functions converting thing back.

The forward DCT is defined as :

$$F(u,v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left[\frac{\pi(2x+1)u}{16}\right] \cos\left[\frac{\pi(2y+1)v}{16}\right]$$

for $u = 0, \ldots, 7$ and $v = 0, \ldots, 7$

$$\text{where } C(k) = \begin{cases} 1/\sqrt{2} & \text{for } k = 0 \\ 1 & \text{otherwise} \end{cases} \quad \text{--(1)}$$

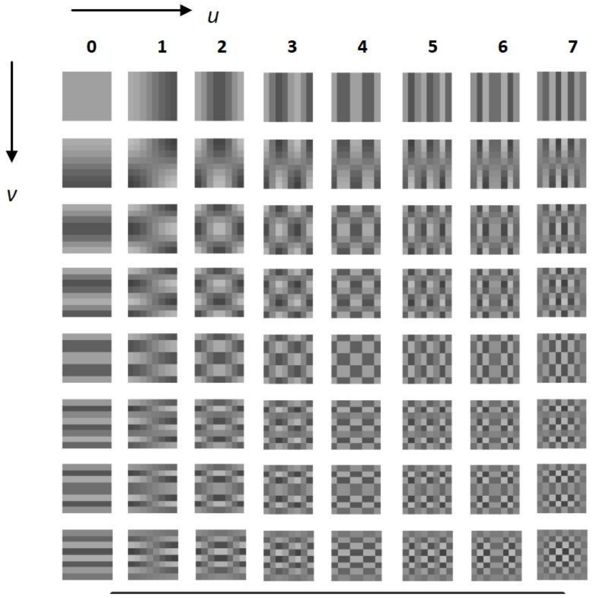


Fig: 8x8 DCT basis

And the inverse DCT is defined as the following equation:

$$f(x,y)=\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)F(u,v)\cos\left[\frac{\pi(2x+1)u}{16}\right]\cos\left[\frac{\pi(2y+1)v}{16}\right]$$

for $x=0,\ldots,7$ and $y=0,\ldots,7$

--(2)

The (u,v) is called the DCT coefficient, and the basis of DCT is:

$$\omega_{x,y}(u,v)=\frac{C(u)C(v)}{4}\cos\left[\frac{\pi(2x+1)u}{16}\right]\cos\left[\frac{\pi(2y+1)v}{16}\right]$$

Now we can write the equation if IDCT:

$$f(x,y)=\sum_{u=0}^{7}\sum_{v=0}^{7}F(u,v)\omega_{x,y}(u,v) \quad \text{for } x=0,\ldots,7 \ \text{ and } \ y=0,\ldots,7$$

--(3)

4. PERFORMANCE EVALUATION METRICS

Two of the error metrics used to compare the various image compression techniques are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error. The mathematical formulae for the two are

$$\text{MSE}=\frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}\left[I(x,y)-I'(x,y)\right]^2 \qquad \text{-(4)}$$

PSNR = 20 * log10 (255 / sqrt (MSE)).
The compression ratio is defined as follows:
CR=n1/n2.

Where *n1* is the data rate of original image and *n2* is that of the encoded bit-stream.
Where I(x,y) is the original image, I'(x,y) is the approximated version (which is actually the decompressed image) and M,N are the dimensions of the images.. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction.

5.Method used for Watermarking :

Our project applied both visible and invisible watermarking technique. And each of them has unique technique and algorithm for watermarking.

Visible watermarking :

In visible watermarking, we used buffered image for putting a visible watermark on the image. First the buffered image loaded the image that we want to put watermark on. And then using Rectangle2D class we write the watermark text on it. And then blurred the font/text. And then the program saves the watermarked image in the directory.

Fig: A demonstration of visible watermarking

Invisible watermarking :

In invisible watermarking, we used the Steganography method to embed a text message into an image file. The program lets you choose an image file from the directory and then lets you choose a text document which has smaller memory size than the image file size. And then the program using steganography method embeds the text document into the image file. The program also lets you retrieve the embedded text document hidden inside the image file by reverse steganography method. The program hides the document / text file and generates a key which only the program knows.

Then when the user wants to retrieve the data it simply extracts the hidden document from the image file by using the key. That's how we managed to do invisible watermarking on an image file.

The program takes the text file and generates an array or bytes. And then the program generates and one dimensional array or bytes which has the text file and the length of the text message in byte. The array has the length of the text document at it's zeroth position in byte. And the rest of the bytes are   text file. The pixels are converted into bytes by a function called intToBytes. And the byte array then gets embedded into the pixel array or bytes by simply changing the last bit of the byte elements of every pixel byte.

As the last bit is least significant bit so by changing the last bit of every pixel data doesn't change the image which could be noticeable by human naked eye. And as we said the program generates a key so it's quiet impossible to find out from which pixel the text documents are being started to hide. So the overall size of the file will increase although the image quality will not change that much as we are just changing the last bit of every pixel byte. But by changing the last bit we are getting a sufficient amount of space for hiding quiet a large document file which can be any kind of signature of the owner or company which owns the image.

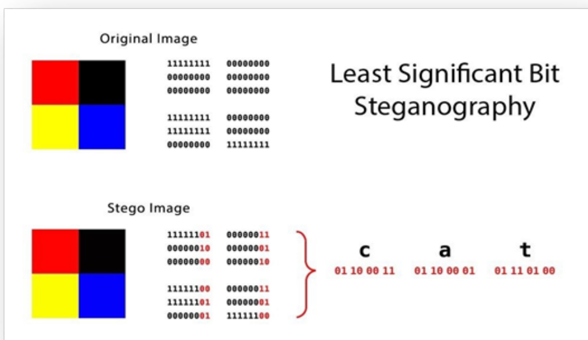


Fig: Steganography procedure

5.Conclusion

Although the image compression is a trade of between compression ratio and peak signal to noise ratio, better and efficient compression-decompression algorithm is yet a demanding in the field. Though extensive research have been taking place in this area, keeping in view the ever increasing need for low bit rate compression methods, scope exists for new methods as well as evolving more efficient algorithms in the existing methods. In our project we didn;t implement any lossy compression methods which could have resulted in compression ratio.We tried to implement DCT until the very end but couldn't quite finish. Also we used steganography instead of digital watermarking due to its complexity. Hopefully we would work on our limitations in the future projects.

References

[1] G.M. Padmaja, C.H. Rajya Lakshmi "Analysis of various image compression techniques", *International journal of Reviews in computing*, April 2012

[2] Abhishek Kaushik, Maneesah Gupta,"Analysis of image compression algorithms", *International journal of engineering research and application*, April 2012.

[3] Woods, R. C. 2008. Digital Image processing. New Delhi: Pearson Pentice Hall, Third Edition, Low price edition, Pages 1-904.

[4] Mamta Sharma, "Compression Using Huffman Coding" *IJCSNS International Journal of Computer Science and Network Security*, VOL.10 No.5, May 2010.

[5] Kity Arora, Manshi Shukla "A Comprehensive Review of Image Compression Techniques", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014