A Study of Image Compression Methods

ANJU
M.Tech(CSE), Hindu College Of Engineering

Abstract: *Image compression is now essential for applications such as transmission and storage in data bases. In this paper we review and discuss about the image compression, need of compression, its principles, and types of compression and various algorithm of image compression. This paper attempts to give a recipe for selecting one of the popular image compression algorithms based on Wavelet, DCT, and VQ. We review and discuss the implementation of these algorithms.*
Initially, the colored image of any resolution is selected by the user which gets converted into grayscale image to be taken as an input image for both the techniques. Finally the Performance analysis of the image compression using self organized feature maps and JPEG 2000 Algorithm.
Keywords: - *Image Compression, Types, SOM, JPEG2000, Algorithms.*

I.   Introduction

Early evidence of image compression suggests that this technique was, in the beginning, most commonly used in the printing, data storage and telecommunications industries. Now-a-days, the digital form of image compression is also being put to work in industries such as fax transmission, satellite remote sensing and high definition television, to name but a few. In certain industries, the archiving of large numbers of images is required. A good example is the health industry, where the constant scanning and/or storage of medical images and documents take place. Image compression offers many benefits here, as information can be stored without placing large loads on system servers [1]. Depending on the type of compression applied, images can be compressed to save storage space, or to send to multiple physicians for examination and these images can uncompress when they are ready to be viewed, retaining the original high quality and detail that medical imagery demands.

Image compression is also useful to any organization which requires the viewing and storing of images to be standardized, such as a chain of retail stores or a federal government agency. In the retail store example, the introduction and placement of new products or the removal of discontinued items can be much more easily completed when all employees receive, view and process images in the same way. Federal government agencies that standardize their image viewing, storage and transmitting processes can eliminate large amounts of time spent in explanation and problem solving. The time they save can then be applied to issues within the organization, such as the improvement of government and employee programs. In the security industry, image compression can greatly increase the efficiency of recording, processing and storage. However, in this application it is imperative to determine whether one compression standard will benefit all areas. For example, in a video networking or closed-circuit television application, several images at different frame rates may be required. Time is also a consideration, as different areas may need to be recorded for various lengths of time. Image resolution and quality also become considerations, as does network bandwidth and the overall security of the system. Museums and galleries consider the quality of reproductions to be of the extreme importance. Image compression, therefore, can be very effectively applied in cases where accurate representations of museum or gallery items are required, such as on a web site. Detailed images which offer short download times and easy viewing benefit all types of visitors, from the student to the discriminating collector. Compressed images can also be used in museum or gallery kiosks for the education of that establishment's visitors. In a library scenario, students and enthusiasts from around the world can view and enjoy a multitude of documents and texts without having to incur traveling or lodging costs to do so. Regardless of industry, image compression has virtually endless benefits wherever improved storage, viewing and transmission of images are required.

The basic idea behind the method of compression is to treat a digital image as an array of numbers i.e., a matrix. Each image consists of a fairly large number of little squares called **pixels** (picture elements). The matrix corresponding to a digital image assigns a whole number to each pixel. For example, in the case of a 256x256 pixel gray scale image, the image is stored as a 256x256 matrix, with each element of the matrix being a whole number ranging from 0 (for black) to 225 (for white). The JPEG compression technique divides an image into 8x8 blocks and assigns a matrix to each block [9].
There are two types of image compression algorithm which can be used to compress the image: Lossless Compression Algorithm & Lossy Compression Algorithm.

1.1 Lossless Compression Technique

In lossless compression techniques, the original image can be perfectly recovered from the compressed (encoded) image. These are also called noiseless since they do not add noise to the signal (image). It is also known as entropy coding since it use statistics/decomposition techniques to eliminate/minimize redundancy. Following techniques are included in lossless compression:

- Run length encoding
- Huffman encoding

1.1.1 Run Length Encoding

Run Length encoding performs lossless data compression which is a very simple compression method used for sequential data. It is very useful in case of repetitive data. It replaces sequences of identical symbols (pixels), called runs by shorter symbols. The run length code for a gray scale image is represented by a sequence $\{V_i , R_i\}$ where V_i is the intensity of pixel and R_i refers to the number of consecutive pixels with the intensity V_i as shown in the figure. If both V_i and R_i are represented by one byte, this span of 12 pixels is coded using eight bytes yielding a compression ratio n of 1: 5

8	8	8	8	8	7	7	7	9	9

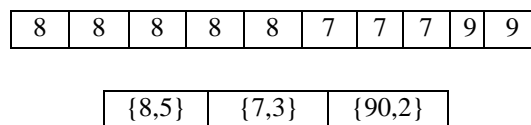| {8,5}	{7,3}	{90,2}

Fig.1: Run –Length Encoding

1.1.2 Huffman Encoding

Huffman encoding based on their statistical occurrence frequencies (probabilities). The pixels in the image are treated as symbols. The symbols that occur more frequently are assigned a smaller number of bits, while the symbols that occur less frequently are assigned a relatively larger number of bits. Huffman code is a prefix code. This means that the (binary) code of any symbol is not the prefix of the code of any other symbol. Most image coding standards use lossy techniques in the earlier stages of compression and use Huffman coding as the final step.

1.2 Lossy compression technique

Lossy schemes provide much higher compression ratios than lossless schemes. Lossy schemes are widely used since the quality of the reconstructed images is adequate for most applications.
Lossy compression techniques includes following schemes:

- Transformation coding
- Vector quantization
- Fractal coding
- Block Truncation Coding
- Subband coding

1.2.1 Transformation Coding

In this coding scheme, transforms such as DFT (Discrete Fourier Transform) and DCT (Discrete Cosine Transform) are used to change the pixels in the original image into frequency domain coefficients (called transform coefficients).These coefficients have several desirable properties. One is the energy compaction property that results in most of the energy of the original data being concentrated in only a few of the significant transform coefficients. This is the basis of achieving the compression. Only those few significant coefficients are selected and the remaining is discarded. The selected coefficients are considered for further quantization and entropy encoding. DCT coding has been the most common approach to transform coding.

1.2.2 Vector Quantization

The basic idea in this technique is to develop a dictionary of fixed-size vectors, called code vectors. This technique is a lossy compression technique. A vector is usually a block of pixel values. A image is then partitioned into non-overlapping blocks called image vectors. Thus, each image is represented by a sequence of indices that can be further entropy coded. After the quantization technique, the Zigzag reordering is done.

Fig.2: Zigzag ordering of JPEG image components

1.2.3 Fractal Coding

The essential of Fractal Coding is to decompose the image into segments by using standard image processing techniques such as color separation, edge detection, and spectrum and texture analysis. Then each segment is looked up in a library of fractals. The library actually contains codes called iterated function system (IFS) codes, which are compact sets of numbers. Using a systematic procedure, a set of codes for a given image are determined, such that when the IFS codes are applied to a suitable set of image blocks yield an image that is a very close approximation of the original. This scheme is highly effective for compressing images that have good regularity and self-similarity.

1.2.4 Block truncation coding

In this, the image is divided into non overlapping blocks of pixels. For each block, threshold and reconstruction values are determined. The threshold is usually the mean of the pixel values in the block. Then a bitmap of the block is derived by replacing all pixels whose values are greater than or equal (less than) to the threshold by a 1 (0). Then for each segment (group of 1s and 0s) in the bitmap, the reconstruction value is determined. This is the average of the values of the corresponding pixels in the original block.

1.2.5 Sub band coding

In this scheme, the image is analyzed to produce the components containing frequencies in well-defined bands, the sub bands. Subsequently, quantization and coding is applied to each of the bands. The advantage is that the quantization and coding well suited for each of the sub bands can be designed separately [4].

Fig.3: Block Diagram of Sub Band Coding

II. Algorithm for Image Compression using JPEG 2000 Standard

The Process of JPEG 2000 Standard is:
1. The Image is broken into 8x8 Blocks of Pixels.
2. Working from Left to Right, Top to Bottom, the DCT is applied to each Block.
3. Each Block is compressed through Quantization.
4. The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.
5. When desired, the image is reconstructed through decompression, a process that uses the Inverse discrete Cosine Transformation (IDCT).

III. Applications of JPEG 2000

1. Consumer applications such as multimedia devices (e.g., digital cameras, personal digital assistants, 3G mobile phones, color facsimile, printers, scanners, etc.)
2. Client/server communication (e.g., the Internet, Image database, video streaming, video server, etc.)
3. Military/surveillance (e.g., HD satellite images, Motion detection, network distribution and storage, etc.)
4. Medical Imaging.
5. Remote sensing
6. High-quality frame-based video recording, editing and storage [7].

IV. SOM

A **self-organizing map** (**SOM**) or **self-organizing feature map** (**SOFM**) is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional, discredited representation of the input space of the training samples, called a **map**. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space.

SOFM learn to classify input vectors according to how they are grouped in the input space. They differ from competitive layers in that neighboring neurons in the SOM learn to recognize neighboring sections of the input space. Thus, self-organizing maps learn both the distribution and topology of the input vectors they are trained on [2]. It consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a regular spacing in a hexagonal or rectangular grid. It describes a mapping from a higher dimensional input space to a lower dimensional map space. The procedure for placing a vector from data space onto the map is to first find the node with the closest weight vector to the vector taken from data space. Once the closest node is located it is assigned the values from the vector taken from the data space.

While it is typical to consider this type of network structure as related to feed forward networks where the nodes are visualized as being attached, this type of architecture is fundamentally different in arrangement and motivation. Useful extensions include using steroidal grids where opposite edges are connected and using large numbers of nodes [5]. It has been shown that while SOM with a small number of nodes behave in a way that is similar to K-means, larger SOM rearrange data in a way that is fundamentally topological in character. It is also common to use the U-Matrix. The U-Matrix value of a particular node is the average distance between the node and its closest neighbors [6]. In a square grid for instance, we might consider the closest 4 or 8, or six nodes in a hexagonal grid.

The principal goal of SOM is to transform an incoming signal pattern of arbitrary dimension into a one- or two- dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion. Fig.4 shows the schematic diagram of a two-dimensional lattice of neurons commonly used as a discrete map. Each neuron in the lattice is fully connected to all source nodes in the input layer [6]. This network represents a feed forward structure with a single computational layer consisting of neurons arranged in row and columns.
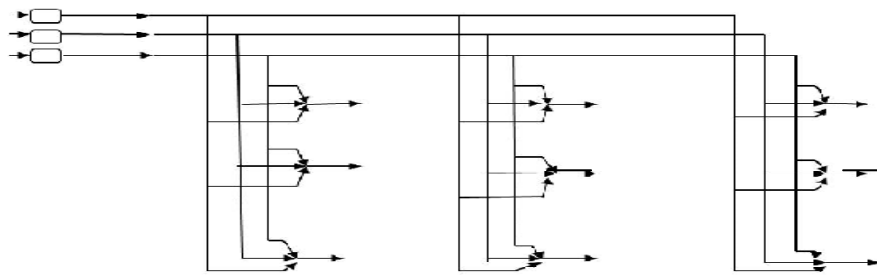
Fig.4: Two- Dimensional lattice of neurons

4.2. Algorithm for SOM

The algorithm responsible for the formation of the SOM proceeds first by initializing the synaptic weights in the network. This can be done by assigning them small values picked from a random number generator. Once the network has been properly initialized, there are three essential processes involved in the formation of the SOM, as summarized as:

1. Competition: For each input pattern, the neurons in the network compute their respective values of a discriminate function. This function provides the basis for competition among neurons. The particular neuron with the largest value of discriminate function is declared winner of the competition.
2. Cooperation: The winning neuron determines the spatial location of a topological neighborhood of excited neurons, providing the basis for cooperation among such neighboring neurons.
3. Synaptic Adaptation: This last mechanism enables the excited neurons to increase their individual values of the discriminated function in relation to the input pattern through suitable arrangements applied to their synaptic weights. The essential parameters of the algorithm are:
    i. A continuous input space of activation patterns that are generated in accordance with a certain probability distribution.
    ii. A topology of the network in the form of a lattice of neurons, which defines a discrete output space.
    iii. A time varying neighborhood function hj,i(x)(n) that is defined around a winning neuron i(x).
    iv. A learning rate parameter ŋ(n) that starts at an initial value ŋ0 and then decreases gradually with time, n, but never goes to zero.

4.3 Learning Vector Quantization (VQ)

A type of neural network consisting of a set of vectors, the position of which are optimized with respect to a given data set. The network consists of an input and output layer, [8] with vectors storing the *connection weights* leading from input to output neurons. The learning method of learning vector quantization is also called "competition learning." For each training pattern cycle an input neuron finds the closest vector, and selects the corresponding output neuron as the "winner neuron." The weights of the connection to the winner are then adapted - either closer to or farther away from the training pattern, based on the class of the neuron. This movement is controlled by a *learning rate* parameter. It states how far the reference vector is moved. Usually the learning rate is decreased in the course of time, so that initial changes are larger than changes made in later epochs of the training process. Learning may be terminated when the positions of the reference vectors hardly change any more.

References

[1.]   Banerjee and A. Halder,"An Efficient Dynamic Image Compression Algorithm based on Block Optimization, Byte Compression and Run-Length Encoding along Y-axis", IEEE Transaction, 2010.
[2.]   Y. H. Dandawate and M.A. Joshi, "Performance analysis of Image Compression using Enhanced Vector Quantizer designed with Self Organizing Feature Maps: The Quality perspective" IEEE transaction, pg. 128-132, 2007.
[3.]   S. Immanuel Alex Pandian and J.Anitha,"A Neural Network Approach for Color Image Compression in Transform Domain", International Journal of Recent Trends in Engineering, Vol 2, No. 2, November 2009, 152.
[4.]   H. G. Lalgudi, A. Bilgin, M. W. Marcellin, and M. S. Nadar, "Compression of Multidimensional Images Using JPEG2000", IEEE Signal processing, vol. 15, pg. 393-396, 2008.
[5.]   Cheng-Fa Tsai, Chen-An Jhuang, Chih-Wei Liu, "Gray Image Compression Using New Hierarchical Self-Organizing Map Technique, IEEE conference, 2008.
[6.]   D. Kumar, C.S. Rai and S. Kumar, "Face Recognition using Self-Organizing Map and Principal Component Analysis", IEEE Transaction, 2005.
[7.]   Gregory K. Wallace, "The JPEG Still Picture Compression Standard", IEEE Transactions on Consumer Electronics, December 1991.
[8.]   S. Anna Durai & E. Anna Saro, "An Improved Image Compression approach with Self-Organizing Feature Maps", GVIP Journal, Volume 6, Issue 2, September, 2006.
[9.]   Sonal, Dinesh Kumar, "A Study of various Image Compression Techniques", IEEE Transaction, 2005.