CHAPTER 3

AN INTRODUCTION TO IMAGE COMPRESSION

3.1 Introduction

The motto of the image compression is to convert the image to a space efficient compressed form so that the data of image is protected as far as possible when decompressing the encoded image. Image compression is an application of data compression that encodes the original image with few bits. The objective of image compression is to reduce the redundancy of the image and to transmit and store data in an efficient form. Compressing an image is significantly different than compressing raw binary data. The general purpose of compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth and storage space.

Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. This is the case when binary data such as executables, documents etc. are compressed. They need to be exactly reproduced when decompressed. On
the other hand, images need not be reproduced 'exactly'. An approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable.

Computer graphics applications, particularly those generating digital photographs and other complex color images, can generate very large file sizes. Issues of storage space, and the need to rapidly transmit image data across networks and over the Internet, has led to the development of a range of image compression techniques in order to reduce the physical size of files. Most compression techniques are independent of specific file formats - indeed, many formats support a number of different compression types. They are an essential part of digital image creation, use, and storage. There are a number of factors to be considered when using compression Algorithms which are particularly suited to specific circumstances; these must be understood if they are to be used effectively. Some, for example, are more efficient at compressing monochrome images, whilst others yield better results with complex color images.

Image compression Algorithms fall into two categories: Lossy compression achieves its effect at the cost of a loss in image quality, by removing some image information [27]. Lossless compression techniques reduce size whilst preserving all of the original image information, and therefore without degrading the quality of the image. Lossy compression
techniques should be treated with caution; if images are repeatedly 
migrated over time between different lossy formats, the image quality will 
be increasingly degraded at each stage. However, in some circumstances 
the use of lossy compression may be required, for example, to enable 
very large volumes of high-quality color images to be managed 
economically. In such circumstances, visually-lossless compression 
should be used, which only removes image information which is invisible 
to the human eye at normal magnification.

Image compression coding is, to store the image into bit-stream as 
compact as possible and to display the decoded image in the monitor as 
exact as possible. In Fig. 3.1 When the encoder receives the original 
image file, the image file will be converted into a series of binary data, 
which is called the bit-stream. The decoder then receives the encoded bit-
stream and decodes it to form the decoded image[37][38]. If the total data 
quantity of the bit-stream is less than the total data quantity of the 
original image, then this is called image compression.

![Diagram of Image Compression Coding]

Fig 3.1 General flow of Image Compression Coding

The compression ratio is defined as follows:

\[
C \cdot R = \frac{n \cdot \frac{1}{n}}{2}
\]

\[\text{--------- 3.1}\]
Where \( n1 \) is the data rate of original image and \( n2 \) is that of the encoded bit-stream.

### 3.1.1 Encoding flow of Image Compression

![Encoding flow of Image Compression](image)

Fig 3.2 Encoding flow of Image Compression

The steps involved in compressing an image are

1. Specifying the Rate and Distortion that is tolerable error for the target image.
2. Dividing the image data into blocks
3. Dividing the available bit rate budget among these blocks such that the distortion is a minimum.
4. Quantize each block separately using the bit allocation information derived in step 3.
5. Encode each block separately using an entropy coder and get the bit stream.

Reconstructing the image from the compressed data is reverse and faster process than compression.

The steps involved in decoding process,
• Read in the quantized data from the file, using an entropy decoder.
• Dequantize the data.
• Rebuild the image.

An image is represented as a two-dimensional array of coefficients, each coefficient representing the brightness value in that point. Most natural images have smooth color variations, with the fine details being represented as sharp edges in between the smooth variations. Technically, the smooth variations in color can be termed as low frequency variations and the sharp variations as high frequency variations[36]. The low frequency components that is, smooth variations constitute the base of an image, and the high frequency components, that is the edges which give the detail add upon them to refine the image, thereby giving a detailed image. Hence, the smooth variations are demanding more importance than the details. Separating the smooth variations and details of the image can be done in many ways. One such way is the decomposition of the image using a Discrete Wavelet Transform.

3.1.2 Reducing the pixels Correlation

The reason for image compression is that the correlation between one pixel and its neighbor pixels is very high, or we can say that the values of one pixel and its adjacent pixels are very similar. Once the
correlation between the pixels is reduced. It can take advantage of the statistical characteristics and the variable length coding theory to reduce the storage quantity. This is the most important part of the image compression Algorithm; there are a lot of relevant processing methods being proposed. The best-known methods are, Predictive Coding, Orthogonal Transform and Sub band Coding.

Predictive Coding is a lossless coding method, which means that the decoded image and the original image have the same value for every corresponding element. Karhunen-Loève Transform (KLT) and Discrete Cosine Transform (DCT) are the two most well-known orthogonal transforms [28]. The DCT-based image compression standard such as JPEG is a lossy coding method that will result in some loss of details and unrecoverable distortion[39]. Sub band coding such as Discrete Wavelet Transform (DWT) is also a lossy coding method. The objective of Sub band coding is to divide the spectrum of one image into the low pass and the high pass components. JPEG 2000 is a 2-dimension DWT based image compression standard.

3.1.3 Quantization

The objective of quantization is to reduce the precision and to achieve higher compression ratio. For instance, the original image uses 8 bits to store one element for every pixel; if we use fewer bits such as 6 bits to save the information of the image, then the storage quantity will
be reduced, and the image can be compressed. The shortcoming of quantization is that it is a lossy operation, which will result into loss of precision and unrecoverable distortion. The image compression standards such as JPEG and JPEG 2000 have their own quantization methods.

Quantization refers to the process of approximating the continuous set of values in the image data with a finite set of values[60]. The input to a quantizer is the original data, and the output is always one among a finite number of levels. The quantizer is a function whose set of output values are discrete, and usually finite [52]. Obviously, this is a process of approximation, and a good quantizer is one which represents the original signal with minimum loss or distortion.

There are two types of quantization - Scalar Quantization and Vector Quantization. In scalar quantization, each input symbol is treated separately in producing the output, while in vector quantization the input symbols are clubbed together in groups called vectors, and processed to give the output. This clubbing of data and treating them as a single unit increases the optimality of the vector quantizer, but at the cost of increased computational complexity. A quantizer can be specified by its input partitions and output levels. If the input range is divided into levels of equal spacing, then the quantizer is termed as a Uniform
Quantizer, and if not, it is termed as a Non-Uniform Quantizer[58]. A uniform quantizer can be easily specified by its lower bound and the step size. Also, implementing a uniform quantizer is easier than a non-uniform quantizer.

Dequantizer is one which receives the output levels of a quantizer and converts them into normal data, by translating each level into a 'reproduction point' in the actual range of data.

The first step in compressing an image is to separate the image data into different blocks. Depending on the importance of the data it contains, each block is allocated a portion of the total bit budget, such that the compressed image has the minimum possible distortion. This procedure is called Bit allocation [59].

3.1.4 Entropy Coding

After the data has been quantized into a finite set of values, it can be encoded using an Entropy Coder, to give additional compression. The main objective of entropy coding is to achieve less average length of the image. Entropy coding assigns code words to the corresponding symbols according to the probability of the symbols. In general, the entropy encoders are used to compress the data by replacing symbols represented by equal-length codes with the code words whose length is
inverse proportional to corresponding probability. By entropy, we mean
the amount of information present in the data, and an entropy coder
encodes the given set of symbols with the minimum number of bits
required to represent them. Two of the most popular entropy coding
schemes are,

- Arithmetic coding
- Huffman coding

Arithmetic coding is a form of entropy encoding used in lossless
data compression. Normally, a string of characters such as the words
"hello there" is represented using a fixed number of bits per character, as
in the ASCII code. When a string is converted to arithmetic encoding,
frequently used characters will be stored with fewer bits and not-so-
frequently occurring characters will be stored with more bits, resulting in
fewer bits used in total. Arithmetic coding differs from other forms of
entropy encoding such as Huffman coding in that rather than separating
the input into component symbols and replacing each with a code,
arithmetic coding encodes the entire message into a single number, a
fraction \( n \) where \( 0.0 \leq n < 1.0 \). Arithmetic coding, is entropy coder
widely used, the only problem is its speed, but compression tends to be
better than Huffman can achieve.
The Algorithm to compute the output number is:

- Low = 0
- High = 1
- Loop. For all the symbols.
  - Range = high - low
  - High = low + range * high range of the symbol being coded
  - Low = low + range * low range of the symbol being coded

Where:

- Range, keeps track of where the next range should be.
- High and low, specify the output number.

Huffman coding is an entropy encoding Algorithm used for lossless data compression. This coding refers to the use of a variable-length code table for encoding a source symbol where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol[36]. Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code that expresses the most common source symbols using shorter strings of bits than are used for less common source symbols.
Huffman coding is based on the frequency of occurrence of a data item that is pixel in images. The principle is to use a lower number of bits to encode the data that occurs more frequently. Codes are stored in a Code Book which may be constructed for each image or a set of images. In all cases the code book plus encoded data must be transmitted to enable decoding. The steps for performing Huffman coding is,

- Scan text to be compressed and tally occurrence of all characters.
- Sort or prioritize characters based on number of occurrences in text.
- Build Huffman code tree based on prioritized list.
- Perform a traversal of tree to determine all code words.
- Scan text again and create new file using the Huffman codes.

Huffman coding is a technique used to compress files for transmission. It uses statistical coding. The more frequently used symbols have shorter code words. It works well for text and fax transmissions and also an application that uses several data structures.

3.2 Size of the Image File

In raster images, Image file size is definitely correlated to the number of pixels in an image and the bits per pixel of the image. Images
can be compressed in various ways. Compression uses an Algorithm that stores an exact representation of the original image in a smaller number of bytes that can be expanded back to its uncompressed form with a corresponding decompression Algorithm. Considering different compressions, it is common for two images of the same number of pixels and color depth to have a very different compressed file size. Taking into consideration exactly the same compression, number of pixels, and color depth for two images, different graphical complexity of the original images may also result in very different file sizes after compression due to the nature of compression Algorithms. With some compression formats, images that are less complex may result in smaller compressed file sizes. This feature sometimes results in a smaller file size for some lossless formats than lossy formats. For example, simple images may be losslessly compressed into a GIF or PNG format and result in a smaller file size than a lossy JPEG format [39]. Vector images, unlike raster images, can be any dimension independent of file size. File size increases only with the addition of more vectors.

**3.3 Compressed Image File Formats**

There are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet. These graphic formats are separated into the two main families of graphics:
raster and vector. In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

**JPEG/JFIF Format**

JPEG (Joint Photographic Experts Group) is a compression method; JPEG-compressed images are typically stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. The JPEG/JFIF file name extension is JPG or JPEG. Every digital camera can save images in the JPEG/JFIF format, which supports 8-bit grayscale images and 24-bit color images (8 bits each for red, green, and blue). JPEG applies lossy compression to images, which can result in a significant reduction of the file size. The amount of compression can be specified, and the amount of compression affects the visual quality of the result. When not too great, the compression does not noticeably detract from the image’s quality, but JPEG files suffer generational degradation when repeatedly edited and saved.

**JPEG 2000 Format**

JPEG (Joint Photographic Expert Group) is a compression standard enabling both lossless and lossy storage. The compression
methods used are different from the ones in standard JFIF/JPEG; they improve quality and compression ratios, but also require more computational power to process. JPEG 2000 also adds features that are missing in JPEG. It is not nearly as common as JPEG, but it is used currently in professional movie editing and distribution.

**Exif Format**

The Exif (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions; it is incorporated in the JPEG-writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing and viewing software. The metadata are recorded for individual images and include such things as camera settings, time and date, shutter speed, exposure, image size, compression, name of camera, color information. When images are viewed or edited by image editing software, all of this image information can be displayed. The actual Exif metadata as such may be carried within different host formats, e.g. TIFF, JFIF (JPEG) or PNG.

**TIFF Format**

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively. TIFFs can be lossy and lossless; some
offer relatively good lossless compression for bi-level (black&white) images. Some digital cameras can save in TIFF format, using the LZW compression Algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business.

**RAW Format**

RAW refers to (Raw Image Formats) that are available on some digital cameras, quite than to a specific format. These formats usually use a lossless or nearly lossless compression, and produce file sizes smaller than the TIFF formats. The raw formats used by most cameras are not standardized or documented. Most camera manufacturers have their own software for decoding or developing their raw file format, but there are also many third party raw file converter applications available that accept raw files from most digital cameras.

**GIF Format**

(Graphics Interchange Format) is limited to an 8-bit palette, or 256 colors. This makes the GIF format suitable for storing graphics with relatively few colors such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression
that is more effective when large areas have a single color, and ineffective for detailed images or dithered images.

**BMP Format**

The BMP file format handles graphics files within the Microsoft Windows OS. Typically, BMP files are uncompressed, hence they are large; the advantage is their simplicity and wide acceptance in Windows programs.

**PNG Format**

The PNG (Portable Network Graphics) file format was created as the free, open-source successor to GIF. The PNG file format supports 8 bit paletted images with optional transparency for all palette colors and 24 bit true color (16 million colors) or 48 bit true color with and without alpha channel - while GIF supports only 256 colors and a single transparent color. Compared to JPEG, PNG excels when the image has large, uniformly colored areas. Thus lossless PNG format is best suited for pictures still under edition - and the lossy formats, like JPEG, are best for the final distribution of photographic images, because in this case JPG files are usually smaller than PNG files.

PNG provides a patent-free replacement for GIF and can also replace many common uses of TIFF. Indexed-color, grayscale, and true color images are supported, plus an optional alpha channel. PNG is
designed to work well in online viewing applications like web browsers so it is fully streamable with a progressive display option. PNG is robust, providing both full file integrity checking and simple detection of common transmission errors. Also, PNG can store gamma and chromaticity data for improved color matching on heterogeneous platforms.

Some programs do not handle PNG gamma correctly, which can cause the images to be saved or displayed darker than they should be. The latter is supported by Mozilla Firefox and Opera and is backwards compatible with PNG.

3.4 Stereo Image Compression Methods

Stereopsis is the ability to perceive three dimensions by merging two slightly different views of the same scene. A stereo image is produced by taking photographs of the same scene from two slightly different positions. The distance between these positions is called the stereo base. Many cameras have the ability to stereoscopically image by sequentially taking two images. Parallax, the apparent displacement of objects in the stereo scene, is achieved by horizontally shifting the camera between photographs with Stereo images provide an improved sense of presence, and have been found to be operationally useful in responsibilities requiring remote manipulation or judgment of spatial relationships[44].
A conventional stereo system with a single left-right pair needs twice the raw data as a monoscopic imaging system. As a result there has been increasing attention given to image compression methods specialized to stereo pairs. In the general concept of stereo image compression method, first compress the reference (left image) image only, after that the block matching Algorithm is applied between left and right image to identify the disparity values and they are encoded separately rather than compressing the whole right image[4].

**General concept of Stereo Image Compression**

![Diagram of General Concept of Stereo Image Compression]

Fig 3.3 General Concept of Stereo Image Compression

**3.4.1 Stereo Image Compression using DCT II**

In Stereo Image Compression using DCT II image transformation, the reference image is left image of stereo pair and that image is
transformed by 2D forward discrete cosine transform (DCT-II). The resultant image matrix is quantized using the quantization matrix and then compressed using Huffman coding. The second part of the encoder involves compressing the right image. Since the two images are very similar to each other, disparity vectors between the two images are estimated with the help of block matching Algorithm. The resulting disparity vectors are displaced pixel values in second image that only compressed into a bit stream using Huffman coding.

3.4.2 Stereo Image Compression using DWT

Stereo pair image compression using Discrete Wavelet Transform (DWT) is being increasingly used for image coding as the DWT can decompose the signals into different sub-bands with both time and frequency information. The main features like progressive image transmission, coding of region of interest and further manipulations in compressed image can be achieved. The separable 2-D wavelet transformation can be implemented by applying a two level decomposition of the 1-D DWT in the horizontal and vertical dimensions respectively [58]. At first step, the right image is estimated from the left image. This step is called disparity estimation/compensation (DS). After the subtraction, the left image and the right error image are transformed. Finally, these images are coded. Stereo images are highly correlated. The left and the right image differ only slightly.
3.4.3 Stereo Image Compression using EBTC

Block Truncation Coding Algorithm is a recent technique used for compression of Still Images. It is extended as EBTC (Extended block truncation coding) Algorithm for stereo pair image compression for the purpose of analysis of compression methods. Block Truncation Coding is a recent technique used for compression of monochrome image data. It is one-bit adaptive moment-preserving quantizer that preserves certain statistical moments of small blocks of the input image in the quantized output.

In block truncation coding Algorithm, it preserves the standard mean and the standard deviation. The statistical overheads Mean and the Standard deviation are to be coded as part of the block. The truncated block of the BTC is the one-bit output of the quantizer for every pixel in the block. The procedure for performing extended block truncation compression for stereo images are, In step one, The left image is divided into non overlapping rectangular regions of size 4 x 4, then select two luminance values to represent each pixel in the block. These values are the mean x and standard deviation o.

By performing the two level quantization on the block, we gain the compression, it is performed by taking x as the threshold value, bit plane is obtained by comparing each pixel value xi with the threshold. If a pixel value is greater than the mean, it is assigned the value "1".
otherwise "0". To reconstruct the image, elements assigned a 0 are replaced with the "a" value and elements assigned to 1 are replaced with the "b" value. To compress the right image: Since the left and right images are similar to each other, the disparity vectors between the two images are estimated. The resulting disparity vectors only are compressed using above procedure.

3.4.4 Block Matching Algorithm

The encoding phase of all the Stereo image pair Algorithm use block matching technique to find the position of object in another block[13]. Therefore the image is divided into a number of rectangular blocks of 8 by 8 pixels. According to the Algorithm used for motion estimation, a block within a search range is compared with the source block[53]. Block matching uses a value called BDM (block distortion measure) to rate the similarity between two blocks. The basic idea is to sum up the differences of the pixel luminances of pixels located at the same position in the two blocks. The blocks which are different in pixel values only undergo the process of compression. The other blocks of the image which are same as reference image are not included in the steps involved in compression of reference image but just copied the bit stream of encoded form of the reference image. So the complexity of compression process is reduced by the block matching Algorithm.
**Disparity Estimation**

Disparity estimation is essentially a correspondence problem. The correspondence between the two images can be determined by either matching features or by operating on or matching of small patches of gray values. Feature matching requires as a preprocessing step the extraction of appropriate features from the images, such as object edges and corners[54]. After obtaining the features, the correspondence problems are first solved for the spatial locations at which the features occur, from which next the full disparity field can be deduced by for instance interpolation or segmentation procedures.

To be able to employ disparity information in a stereoscopic pair in image processing applications, the relation between the contents of the left view image and the right view image has to be established, yielding the disparity field. The disparity field indicates for each point in the left view image the relative shift of the corresponding point in the right view image, and vice versa. Since some parts of one view image may not be visible in the alternate view image due to occlusion, not all points in the image pair can be assigned a disparity vector.

Feature-based disparity estimation is especially useful in the analysis of scenes for robot vision applications. Disparity field estimation by operating directly on the image gray value information is
not unlike the problem of motion estimation. The first difference is that disparity vectors are approximately horizontally oriented[55]. Deviations from the horizontal orientation are caused by the convergence of the camera axes and by differences between the camera optics. Usually vertical disparity components are either ignored or rectified for. A second difference is that disparity vectors can take on a much larger range of values within a single image pair. Furthermore the disparity field may have large discontinuities associated with objects neighboring in the planar projection but having a very much different depth. In those regions of the stereoscopic image pair where one finds large discontinuities in the disparity field due to abrupt depth changes, large regions of occlusion will be present. Estimation methods for disparity fields must therefore be able not only to find the correspondence between information in the left and right view images, but must also be able to perceive and handle discontinuities and super nature of virtual reality beyond the range of our ordinary knowledge.

Most disparity estimation Algorithms used in stereoscopic communications rely on matching small patches of gray values from one view to the gray values in the alternate view. The matching of this small patch is not carried out in the entire alternate image, but only within a relatively small search region to limit the computational complexity. Standard methods typically use a rectangular match block of relatively
small size 8x8 pixels[56]. The relative horizontal shift between a match block and the block within the search region of the alternate image that results in the smallest value of a criterion function used, is then assigned as disparity vector to the center of that match block. Often used criterion functions are the sum of squares and the sum of the absolutes values of the differences between the gray values in the match block and the block being considered in the search region. If the left image is assumed as the anchor frame B_m, the problem is to determine a block in the target frame i.e. the left frame where the difference between the two blocks is minimum.

Fig 3.4 Disparity Estimation
The size of a macro block was taken as 16x16 pixels and each block in the anchor frame was compared with each block in the target frame. The block with the minimum error was calculated by the equation

\[ \mathbb{E}_{DFD} (d_m) = \sum |\Psi_2 (x+d_m) - \Psi_1 (x)|^p \]  

---------- 3.2

An operation is considered as one subtraction, one addition or calculating one absolute difference, the total number of operations are.

Calculating the Minimum Absolute Difference for each block is \( N^2 \)

Calculating the disparity for one block is \((2R+1)^2N^2\) where, the block size is assumed to be \( NxN \) pixels and the search size is \( \pm R \) pixels.

The above procedure is carried out for all pixels, first matching the blocks from the left view image to the right view image, then vice versa. From the combination of the two resulting disparity fields and the values of the criterion function, the final disparity field is computed, and occluding areas in the stereoscopic image pair are detected. For instance, one way of detecting occlusions is a local abrupt increase of the criterion function, indicating that no acceptable correspondence between the two image pairs could be found locally.