IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. 2 (Jan – Feb. 2015), PP 06-11
www.iosrjournals.org

Performance Analysis of Compression Techniques Using SVD, BTC, DCT and GP

K. C. Chandra Sekaran[1], Dr. K. Kuppusamy[2]
[1]*Department of Computer Science, Alagappa Govt. Arts College, Karaikudi, Tamilnadu, India*
[2]*Department of Computer Science and Engineering, Alagappa University, Karaikudi,, Tamilnadu , India*

Abstract: *Digital image compression techniques minimize the size in bytes of a graphics file without degrading the quality of the image to an acceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. The proposed paper summarizes the different compression techniques such as Singular Value Decomposition, Block Truncation Coding, Discrete Cosine Transform and Gaussian Pyramid on the basis of Peak Signal to Noise Ratio, Mean Squared Error, Bit Rate to evaluate the image quality for both gray and RGB images. The comparison of these compression techniques are classified according to different biometric images.*

Keywords: *Biometric, Block Truncation Coding, Discrete Cosine Transform, threshold, Gaussian Pyramid, Singular Value Decomposition, Compression*.

I. Introduction

A biometric verification system is designed to verify or recognize the identity of a living person on the basis of his/her physiological characters, such as face, fingerprint and iris or some other aspects of behavior such as handwriting or keystroke pattern. The need for reliable identification of interacting users is obvious. The biometrics verification technique acts as an efficient method and has wide applications in the areas of information retrieval, automatic banking and control of access to security areas, buildings and so on. When lot of image has to be stored memory required is more. Image compression addresses the problem of reducing the amount of data required to represent a digital image called the redundant data. The underlying basis of the reduction process is the removal of redundant data.

Compression on digital images [1] refers to reducing the quantity of data used to represent an image without excessively reducing the quality of the original data. The main purpose of image compression is to reduce the redundancy and irrelevancy present in the image, so that it can be stored and transferred efficiently. The compressed image is represented by less number of bits compared to original. Hence the required storage size will be reduced, consequently maximum images can be stored and it can be transferred in faster way to save the time transmission bandwidth.

Depending on the compression techniques, the image can be reconstructed with and without perceptual loss [2]. In lossless compression, the reconstructed image after compression is numerically identical to the original image. In lossy compression scheme, the reconstructed image contains degradation relative to the original.

Lossy technique causes image quality degradation in each compression or decompression step. In general, lossy techniques provide for greater compression ratios than lossless techniques, i.e., lossless compression gives good quality of compressed images, but yields only less compression whereas the lossy compression techniques lead to loss of data with higher compression ratio.

Image compression [3] also reduces the time required for images to be sent through internet and intranet. There are several different methods to compress an image file which has its own advantages and disadvantages. In this proposed paper we made a comparative analysis of Singular Value Decomposition, Block Truncation Coding, Discrete Cosine Transform and Gaussian Pyramid techniques based on different performance measure such as Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE) and Compression Ratio (CR).

II. Singular Value Decomposition

The singular value decomposition (SVD) [4] is a highlight of linear algebra technique used to solve many mathematical problems. It plays an interesting, fundamental role in many different applications like digital image processing data compression, signal processing and pattern analysis. The beauty of SVD with its digital applications is that it provides a robust method of storing large images as smaller. This is accomplished by reproducing the original image with each succeeding non zero singular value. Furthermore, to reduce storage size even further, one may approximate a 'good enough' image with using even fewer singular values.

SVD is able to efficiently represent the intrinsic algebraic properties of an image where singular values correspond to the brightness of the image and singular vectors reflect geometry characteristics of the image. An image matrix has many small singular values compared with the first singular value. Even ignoring these small singular values in the reconstruction of the image does not affect the quality of the reconstructed image.

Any image can be considered as a square matrix without loss of generality. So SVD technique can be applied to any kind of images. If it is a grey scale image the matrix values are considered as intensity values and it could be modified directly or changes could be done after transforming images into frequency domain.

SVD method [5] reproduces most photographic images well and allows a significant storage reduction. The SVD is based on decomposing a matrix into two matrices U and V and a vector Σ containing scale factors called singular values. This decomposition of a matrix A is expressed as

$$A = U \sum V^T$$

Each singular value in Σ corresponds to a single two-dimensional image built from a single column in U and a single row in V. The reconstructed image is the sum of each partial image scaled by the corresponding singular value in Σ.

The key of compressing an image is recognizing that the smallest singular values and their corresponding images should not significantly contribute to the final image. By ignoring the smallest singular values multiply the original image should be accurately reconstructed from a data set much smaller than that of the original image. If only a few singular values are needed, a small fraction of the original U and V matrices is needed to reconstruct the image and the storage cost id cut significantly.

The reconstruction of compressed images while ignoring small singular values is illustrated by decomposing a photograph with the singular value decomposition and then reconstructing it with the largest singular values.

III. Block Truncation Coding

Block Truncation Coding (BTC) is one of the simple and fast compression methods and was proposed by Delp and Mitchell [6] in 1979 for the grayscale images. BTC is a lossy compression method. It works by dividing the image into small sub images and then reducing the number of gray levels in each block.

A quantizer that adapts to the local image statistics performs this reduction. The BTC method preserves the block mean and standard deviation. In the encoding procedure of BTC image is first partitioned into a set of non-overlapping blocks.

The following steps involves in BTC algorithm

Step 1: The image is sub-divided into non-overlapping rectangular regions of size m x m.

Step 2: For a two level quantizer of BTC are mean x and standard deviation σ values to represent each pixel in the block.

$$Mean\ x = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i} \qquad (1)$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x - mean(x)\right)^2} \qquad (2)$$

Where x_i represents the i^{th} pixel value of the image block and n is the total number of pixels in that block.

Step 3: Taking the block mean as the threshold value, a two level bit plane is obtained by comparing the pixel value x_i with the threshold.

If x_i is less than the threshold then 0 represents the pixel otherwise by 1. By this process, each binary block is reduced to a bit plane. For example, a block of 4 x 4 pixels will give a 32 bit compressed data amounting to 2 bits per pixel (bpp).

Step 4: In the decoder an image block is reconstructed by replacing 1's in the bit plane with H and the 0's with L, which are given by

$$H = mean(x) + \sigma\sqrt{\frac{p}{q}} \qquad (3)$$

$$L = mean(x) - \sigma\sqrt{\frac{q}{p}} \qquad (4)$$

Where p and q are the number of 0's and 1's in the compressed bit plane respectively. Thus it achieves 2 bits per pixel (bpp) with low computational complexity to vector quantization and transform coding. It was quite nature to extend BTC to multi-spectrum images such as color images.

IV. Discrete Cosine Transform

The Discrete Cosine Transform (DCT) has been applied extensively to the area of image compression. It has excellent energy-compaction properties and as a result has been chosen as the basis for the Joint Photography Experts' Group (JPEG) still picture compression standard. DCT is an example of transform coding. The DCT coefficients are all real numbers and Inverse Discrete Cosine Transform (IDCT) can be used to retrieve the image from its transform representation. DCT is simple when JPEG used for higher compression ratio the noticeable blocking artifacts across the block boundaries cannot be neglected. The DCT can be quickly calculated and is best for images with smooth edges.

DCT is used to transform a signal from the spatial domain into the frequency domain. The DCT represents an image as a sum of sinusoids of varying magnitudes and frequencies. DCT has the property that, for a typical image most of the visually significant information about an image is concentrated in just few coefficients of DCT. After the computation of DCT coefficients, they are normalized according to a quantization table with different scales provided by the JPEG standard. Selection of quantization table affects the entropy and compression ratio.

The value of quantization is inversely proportional to quality of reconstructed image, better mean square error and better compression ratio. In a lossy compression technique during quantization, that the most important frequencies are used to retrieve the image in decomposition process. After quantization, quantized coefficients are rearranged in a zigzag order for further compressed by an efficient lossy coding algorithm.

DCT has the ability to pack most information in fewest coefficients and also it minimizes the block like appearance called blocking artifact that results when boundaries between sub-images become visible. The two dimensional DCT is defined to be Y = CTXC, where X is an N x N image block, Y contains the N x N DCT coefficients, and C is an N x n matrix defined as

$$\mathrm{Cmn} = \mathrm{Kn}\cos\left[\frac{(2\mathrm{m}+1)\mathrm{n}\pi}{2\mathrm{N}}\right] \qquad (5)$$

Where

$$\mathrm{Kn} = \sqrt{\frac{1}{N}}$$

if n=0, $\sqrt{\frac{2}{N}}$ Otherwise, Where m, n = 0, 1, …….. (N - 1).

The signal in the frequency domain contains the same information as that in the spatial domain. The order of values obtained by applying the DCT is coincidentally from lowest to highest frequency. This feature and the psychological observation that the human eye and ear are less sensitive to recognizing the higher order frequencies leads to the possibility of compressing a spatial signal by transforming it to the frequency domain and dropping high-order values and keeping low-lower ones. When reconstructing the signal and transforming it back to the spatial domain, the results are remarkably similar to the original signal.

V. Gaussian Pyramid (Gp) Method

Pyramid compression techniques are applied to images. With two-dimensional images the pyramid is split into several layers with each layer being of a fraction of the original images resolution. One such algorithm works by taking the original image and passing a filter over it, such as a Gaussian blur. Other such methods can include scaling down the original image to a quarter of its original size and then scaling it up again to the original size using various interpolation methods. Using the pyramid coding scheme [7], we decompose the original image into several sub-images depending on the signal characteristics. The initial step in pyramid coding is to low-pass filter the original image GP0 to obtain image GP1. Actually GP1 is a reduced version of GP0 in that both resolution and sample density are decreased. In a similar way, we form GP2 as a reduced version of GP1 and so on. Filtering is performed by a procedure equivalent to convolution with one of a family of local, symmetric weighing functions.The pyramid scheme codes an input image in a multi-resolution representation in the same way as the generation of sub-images of various scales. Here, how resolution sub-images GPk are created by passing GPk-1 through a low-pass filter H. In the encoder scheme [2], we transmit sub-images {L0, L1, LK, GPk+1} obtained by

$$\begin{aligned} &L_0=GP_0-GP_{1i}\\ &L_1=GP_1-GP_{2i}\\ &\cdots\\ &\cdots \qquad (6)\\ &\cdots \end{aligned}$$

$$L_k = GP_k - GP_{(k+1)i}$$
$$GP_{k+1}$$

where L_k is the difference sub-image at the k^{th} level, GP_k is the low-resolution sub-images of the kth level and GP_{ki} is the interpolated version of GP_k (using filter F)

In the decoding scheme, we reverse the equation (6) to get the original signal GP_0. The Pyramid representation has been introduced in the literature for coding purposes as it was shown to be a complete representation. Perfect reconstruction is guaranteed if there is no quantization of the transmitted data, regardless of the choice of filters H and F. Suppose the image is represented initially by the array g_0 which contains C columns and R rows of pixels. Each pixel represents the light intensity at the corresponding image point by an integer I between 0 and k-1. This image becomes the bottom or zero level of the Gaussian Pyramid. Pyramid level 1 contains image g_1, which is reduced version of g_0.

Each value within level 1 is computed as a weighted average of values in level 0 within a 2-by-2 window. Each value within level 2, representing g_2, is then obtained from values within level 1 by applying the same portion of weights. A graphical representation of this process in one dimension is given is fig. 1. The size of the weighting function is not critical. We have selected the 2-by-2 pattern because it provides adequate filtering at low computational cost.

Each row of dots represents nodes within a level of the pyramid. The value of each node in the zero level is just the gray level of a corresponding image pixel. The value of each node in a high level is the weighted average of node values in the next lower level. Note that node spacing doubles from level to level, while the same weighting pattern or "generating kernel" is used to generate all levels.

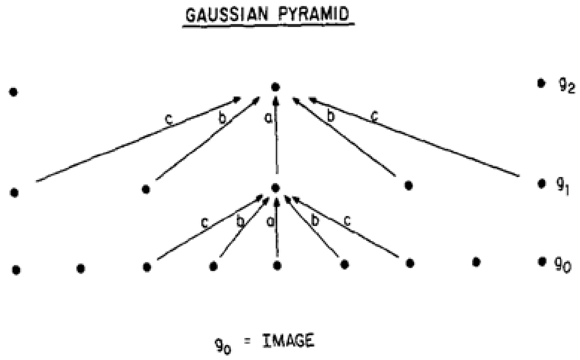

Figure 1: Gaussian Pyramid

The following steps involves in GP algorithm

Step 1: Accept the input image A can be RGB or GRAY scale

Step 2: If given image is GRAY image

Step 3: Image compression is done by low pass filter the image using convolution Fast Fourier Transform [if image is RGB, low pass filter the image using convolution FFT for red, green and blue separately]

Step 4: Downsampling the image and save image in outfile

Step 5: Image reconstruction is by upsampling the image

Step 6: Low pass filter the image with the scaling factor using convolution Fast Fourier Transform.

Input: Infile is the input file name, Level is a scaling factor (positive integer), and Outfile is the output file name

Output: Compressed file is the outfile.

VI. Experimental Results

Experiments are conducted on UPOL[8] iris image database provided by University of Olomue, FVC2000 [9] fingerprint database and COEP [10] palmprint database provided by College Of Engineering Pune using Matlab 7.0 on an Intel Pentium IV 3.0 GHz processor with 512MB memory. Different compression techniques on biometric images are performed and compared based on SNR, PSNR, MSE and Compression Ratio in percentage for both RGB and gray images.

6.1 Images

A digital image is basically a 2-dimensional array of pixels. Images form the significant part of data, particularly in remote sensing, biomedical and video conferencing applications. A biometric verification system is designed to verify or recognize the identity of a living person on the basis of his/her physiological characters, such as face, fingerprint and iris or some other aspects of behavior such as handwriting or keystroke pattern. The biometric verification technique acts as an efficient method and has wide applications in the areas of information retrieval, automatic banking and control of access to security areas, buildings and so on. When lot of image has to be stored, memory required is more. Hence image compression is used to reduce the amount of memory to store an image without much affecting the quality of an image.

6.2 Performance Measures

Once image compression is designed and implemented it is important to evaluate the performance. This evaluation should be done in such a way to be able to compare results against other image compression techniques. The compression efficiency is measured by using the CR. The quality of the image is analyzed by measuring PSNR and MSE.

6.2.1. Compression Ratio (CR)

The performance of image compression can be specified in terms of compression efficiency is measured by the compression ratio or by the bit rate. Compression ratio is the ratio of the size of the original image to the size of the compressed image.

CR=Size of original image / Size of the compressed image

If CR > 1, it is a positive compression.

If CR < 1, it means a negative compression.

Whenever this ratio is large, it indicates that the compression is better otherwise the compression is weak.

6.2.2. Bits Per Pixel / Bit Rate (BR)

The bit rate is the number of bits per pixel required by the compressed image. Let b be the number of bits per pixel (bit depth) of the uncompressed image, CR the compression ratio and BR the bit rate. The bit rate ratio is given by BR = b / CR.

6.2.3. The Mean Squared Error (MSE)

MSE indicate the average difference of the original image data and reconstructed data and results the level of distortion in equation (7). The MSE between the original data and reconstructed data is

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[f(i,j) - f'(i,j)]^2 \quad \text{.... 7}$$

where m and n are the row and column size of the image.

6.2.4 Peak Signal to Noise Ratio (PSNR)

Peak Signal to noise is one of the quantitative measures for image quality evaluation which is based on the Mean Square Error (MSE) of the reconstructed image. PSNR is expressed by equation (8).

$$\mathrm{PSNR} = 10\log\left(\frac{\mathrm{MAX}_i^2}{\mathrm{MSE}}\right) \quad \text{.... 8}$$

where MAXi is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255. In general, when samples are represented using GPC with B bits per sample, MAXi is 2B-1.

For color images with RGB values per pixel, the definition of PSNR is the same except the MSE [11] is the sum over all the squared value differences divided by image size and three. In this case, the large results mean that there is a small noise in the compression system image quality of the reconstructed and image is better. When value of PSNR [12] is small, it means that the compression performance is weak.

Table 1 shows the various compression techniques with uncompressed images and compressed images size in kilobytes. Table 2 represents the performance summary of different compression techniques.

Table 1: Compression Technique Vs Compression Ratio

Images	Compression Technique	Uncompressed Image (KB)	Compressed Image (KB)
Iris [6] (Gray)	BTC	13.9	14.6
	DCT	13.9	2.35
	GP	13.9	5.27
	SVD	13.9	5.64
Finger Prints [7] (Gray)	BTC	12.5	10.7
	DCT	12.5	1.7
	GP	12.5	5.32
	SVD	12.5	4.57

Palm Prints [8] (RGB)	BTC	8.43	8.13
	DCT	8.43	2.18
	GP	8.43	4.55
	SVD	8.43	4.76

Table 2: Comparison Of Compression Techniques

Images	Compression Technique	SNR	PSNR	MSE	BPP	CR
Iris (Gray)	SVD	15.89	19.37	746.94	0.7544	2.09
	BTC	29.40	32.91	33.30	1.5583	1.03
	DCT	27.84	31.13	50.16	1.5429	1.15
	GP	17.75	21.25	487.12	0.7513	2.08
Finger Prints (Gray)	SVD	13.39	19.03	811.68	0.6571	2.09
	BTC	20.21	25.63	178.03	1.196	2.74
	DCT	13.68	19.10	799.42	0.1776	4.22
	GP	15.08	20.49	579.62	0.488	5.73
Palm Prints (RGB)	SVD	10.61	23.27	305.95	0.2003	1.77
	BTC	17.42	30.26	61.15	0.1869	1.04
	DCT	4.88	21.78	431.08	0.1165	2.98
	GP	17.77	20.44	58.75	0.1914	1.85

VII. Conclusion

This paper demonstrates the potential of various image compression techniques with respect to bpp, PSNR, MSE and compression ratio. When compared to bpp for gray images GP is the best and for RGB images DCT is the best. When PSNR values are compared between various techniques SVD performs well for gray images and GP for RGB images. SVD gives better results without losing more information of gray images and gives less compression ratio. Thus on comparing the various methods with respect to the various standards and various images, GP is found to be good for RGB images and SVD is found to be good for gray images.

References

[1]. Pardeep Singh, etal., A Comparative Study: Block Truncation Coding, Wavelet, Embedded Zerotree and Fractal Image Compression on Color Image, Vol 3, Issue-2, pp.10-21, 2012.
[2]. David Solomon, Data compression The Complete Reference 2nd edition, Springer 2001, Newyork.
[3]. Kiran Bindu, etal., A Comparative Study of Image Compression Algorithms, IJRCS, Vol 2, Issue 5,pp 37-42,2012.
[4]. V. CKloma, The singular value decomposition: its computation and some applications, IEEE Transactions in automatic control, Vol. 25, pp. 164-176, 1980.
[5]. Mulcahy, Colm & Rossi, John: Atlast: A Fresh Approach to Singular Value Decomposition.
[6]. Delp E.J, O.R. Mitchell,"Image Compression using Block Truncation Coding", IEEE Trans. Communications, Vol. 27 pp 1335-1342, September 1979.
[7]. P.J.Burt and E.A.Adelson, The Laplacian Pyramid as a Compact Image Code, IEEE Transactions on Communications, Vol. COM-31,pp. 532-50,1983.
[8]. Michal Dobes and Libor Machala, UPOL Iris Image Database, 2004, http://phoenix.inf.upol.cz/iris/.
[9]. FVC2002. http://bias.csr.unibo.it/fvc2002/
[10]. COEP.http://www.coep.org.in/
[11]. Afshan Mulla, Namrata Gunjikar and Radhika Naik, Comparison of Different Image Compression Techniques, IJCA, Vol 70, pp 7-11, 2013.
[12]. Sachin Dhawan, A Review of Image Compression and Comparison of its Algorithms, IJECT, Vol 2, Issue 1, pp.22-26, 2011.