

Comparison of different Fingerprint Compression Techniques

Ms.Mansi Kambli¹ and Ms.Shalini Bhatia²

Thadomal Shahani Engineering College^{1,2}

Email:mansikambli@gmail.com¹

Email: shalini.tsec@gmail.com²

ABSTRACT

The important features of wavelet transform and different methods in compression of fingerprint images have been implemented. Image quality is measured objectively using peak signal to noise ratio (PSNR) and mean square error (MSE).A comparative study using discrete cosine transform based Joint Photographic Experts Group(JPEG) standard , wavelet based basic Set Partitioning in Hierarchical trees(SPIHT) and Modified SPIHT is done. The comparison shows that Modified SPIHT offers better compression than basic SPIHT and JPEG. The results will help application developers to choose a good wavelet compression system for their applications.

KEYWORDS

SPIHT, JPEG, Modified SPIHT, Run Length Encoding, Quality Measures

1. INTRODUCTION

In this era of technology the most important factor is storage space. The transmission of images consumes a lot of space. Image compression is one of key techniques in solving this problem. Image compression exploits redundancy to achieve reduction in the actual amount of data with or without quality information loss according to certain rules through transform and combination. Many image compression algorithms have been in practice, such as DPCM, JPEG, SPIHT and JPEG2000, etc. Even though there are many applications for image compression little work has been done in the progressive retrieval of images. The algorithm should concentrate on this aspect, while at the same time, minimizing the loss of information. The algorithm should provide efficient and progressive compression and decompression for gray-scale and color bitmap (.bmp) images.. The application provides efficient and flexible image compression for bitmap (.bmp) images with variable compression ratios. The application finds use in transmission of images, storage of images as well as image mining.

The overall goal of compression is to represent an image with the smallest possible no of bits, thereby speeding transmission and minimizing storage requirements. Compression is of two types-lossless and lossy. In lossless compression schemes the reconstructed image after compression is numerically identical to the original image. However lossless compression can achieve a modest amount of compression. Lossy schemes though it does not permit perfect reconstruction of the original image can provide satisfactorily quality at a fraction of the original bit rate.

Large volumes of fingerprints are collected and stored every day in a wide range of applications, including forensics, access control etc., and are evident from the database of Federal Bureau of Investigation (FBI) which contains more than 70 million fingerprints. An automatic recognition of people based on fingerprints requires that the input fingerprint be matched with candidates within a large number of fingerprints. Since large volume of data in a database consumes more amount of memory, the information contained in fingerprints must, therefore, be compressed by extracting only visible elements. Fingerprint images exhibit characteristic high energy in certain high frequency bands resulting from the ridge-valley pattern and other structures. To account for this property, the Wavelet standard for lossy fingerprint compression has been proposed in this paper [4].

2. RELATED WORK

2.1. JPEG

JPEG stands for Joint Photographic Experts Group. 'Joint Photographic Experts Group' or JPEG standard has been established by ISO (International Standards Organization) and IEC (International Electro-Technical Commission). The performance of these coders generally degrades at low bit-rates mainly because of the underlying block-based Discrete Cosine Transform (DCT) scheme. The JPEG standard specifies three modes namely sequential, progressive, and hierarchical for lossy encoding, and one mode of lossless encoding.

It works well on photographic images. It is a lossy technique. The DCT-based encoder can be thought of as essentially compression of a stream of 8x8 blocks of image samples. Each 8x8 block makes its way through each processing step, and yields output in compressed form into the data stream. Because adjacent image pixels are highly correlated, the 'forward' DCT (FDCT) processing step lays the foundation for achieving data compression by concentrating most of the signal in the lower spatial frequencies. For a typical 8x8 sample block from a typical source image, most of the spatial frequencies have zero or near-zero amplitude and need not be encoded. In principle, the DCT introduces no loss to the source image samples; it merely transforms them to a domain in which they can be more efficiently encoded.

After output from the FDCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a carefully designed 64-element Quantization Table (QT). At the decoder, the quantized values are multiplied by the corresponding QT elements to recover the original unquantized values. After quantization, all of the quantized coefficients are ordered into the "zigzag" sequence. This ordering helps to facilitate entropy encoding by placing low-frequency non-zero coefficients before high-frequency coefficients. The DC coefficient, which contains a significant fraction of the total image energy, is differentially encoded.

Entropy Coding (EC) achieves additional compression lossless by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. The JPEG proposal specifies both Huffman coding and arithmetic coding. The baseline sequential codec uses Huffman coding, but codec's with both methods are specified for all modes of operation. Arithmetic coding, though more complex, normally achieves 5-10% better compression than Huffman coding.

The use of uniformly sized blocks simplified the compression system, but it does not take into account the irregular shapes within the real images as in fingerprint images. Degradation occurs which is known as blocking effect and it depends on the block size. A larger block leads to more efficient coding but requires more computational power. Image distortion is less annoying for small than for large DCT blocks. Therefore more existing systems use blocks of 8x8 or 16x16 pixels as a compromise between coding efficiency and image quality[6].

Steps:

1. Divide the image into 8x8 sub images.
2. Shift the gray-levels in the range [-128, 127].
3. Apply DCT on the partitioned image (64 coefficients will be obtained: 1 DC coefficient and 63 AC coefficients).
4. Quantize the coefficients and the less significant coefficients are set to zero.
5. Order the coefficients using zigzag ordering and the coefficients obtained are in order of increasing frequency.

2.2. Wavelet based compression techniques

Despite all the advantages of JPEG compression schemes based on DCT namely simplicity, satisfactory performance, and availability of special purpose hardware for implementation; there are shortcomings. Since the input image needs to be "blocked," correlation across the block boundaries is not eliminated. This results in noticeable and annoying "blocking artifacts" particularly at low bit rates.

Over the past several years, the wavelet transform has gained widespread acceptance in signal processing in general, and in image compression research in particular. In many applications wavelet-based schemes (also referred as sub band coding) outperform other coding schemes like the one based on DCT. Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes at higher compression avoid blocking artifacts. Wavelet-based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. Because of their inherent multiresolution nature, wavelet coding schemes are especially suitable for applications where scalability and tolerable degradation are important.

2.2.1. Basic SPIHT

The SPIHT algorithm operates on a wavelet-transformed image with equal length and width of an integer power of 2. It encodes the wavelet coefficients in a way that uses a hierarchical organization of the coefficients. This encoding sends high-order bits of coefficients before low-order bits. The SPIHT algorithm only requires anywhere from 1 to $\log_2 N$ steps of the wavelet transform. Energy is concentrated in the coarser approximations (that is, those coefficients tend to have a larger magnitude) and there is a spatial self-similarity between the parent and child pixels that suggests that an encoding scheme that moves from the parent to the child will exhibit

decreasing coefficient magnitudes. current magnitude threshold. It is a very fast coding/decoding (nearly symmetric) algorithm [5].

Important Terms

SPIHT maintains three lists

List of Significant Pixels (LSP)

List of Insignificant Pixels (LIP)

List of Insignificant Sets (LIS)

Type A entry: $D(i,j)$

Type B entry: $L(i,j)$

What is significant?

For a pixel: $|c(i,j)| \geq 2^n \Rightarrow$ Significant

For a set S: $\max |c(i,j)| \geq 2^n \Rightarrow$ Significant

Where n is the number of bits of the largest coefficient

The SPIHT Algorithm

1. Initialization
2. Sorting Pass
3. Refinement Pass
4. Quantization-step update

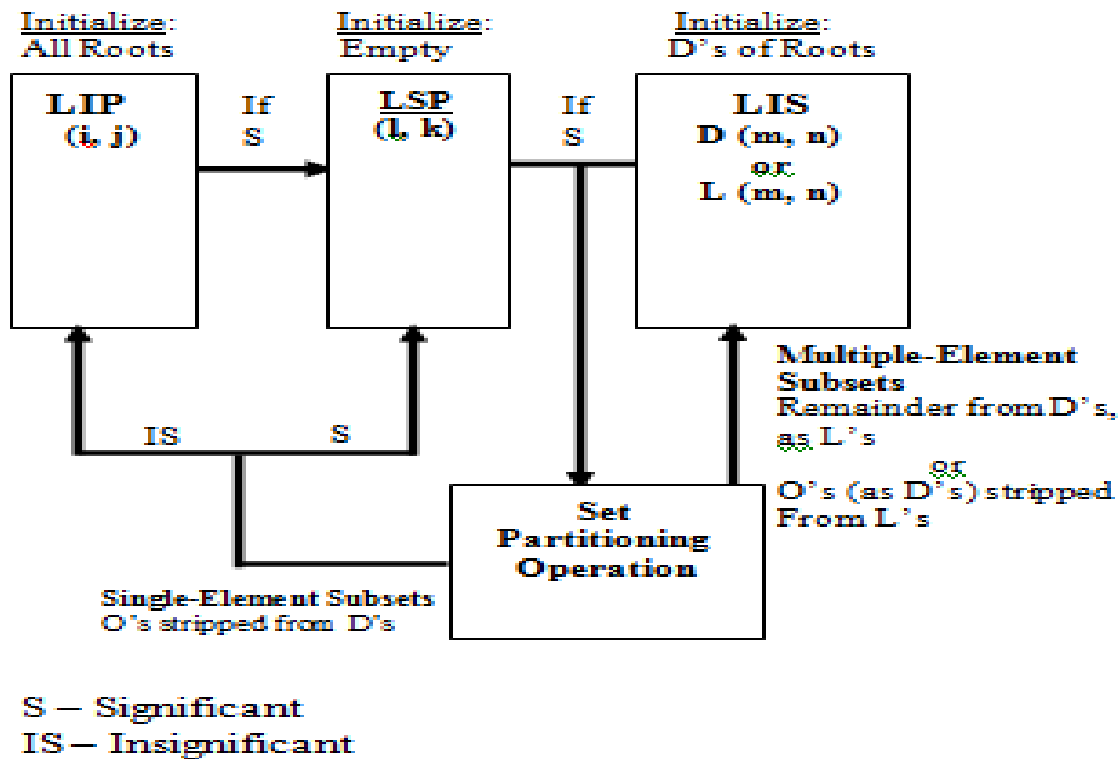


Figure 2.1: SPIHT algorithm

In Figure 2.1 the term significant pixel is used to indicate that the magnitude of a pixel exceeds or equals the current threshold. An insignificant pixel is a pixel whose magnitude is less than the current threshold. An insignificant set can be one of two types of sets. The set H contains all the pixels in the last level of the wavelet transform that was performed, including the coarse and detail coefficients.

$O(i,j)$: set of coordinates of all offspring of node (i,j) .

$D(i,j)$: set of coordinates of all descendants of node (i,j) .

$H(i,j)$: set of all tree roots (nodes in the highest pyramid level).

$L(i,j)$: $D(i,j) - O(i,j)$ (all descendants except the offspring[5]).

Initialization

It initializes the value of n for testing significance of pixels and constructing significance map. The LSP is set as an empty list. The LIS is initialized to maintain all pixels in the low pass sub band that have descendants and hence act as roots of spatial trees. All these pixels are assigned to be of type A. LIP is initializing to contain all pixels in low pass pixels.

1. $n = \lceil \log_2(\max\{c(i,j)\}) \rceil$ where $c(i,j)$ is the coefficient at position (i,j) in the image.
2. LIP = All elements in H

3. LSP = Empty

4. LIS = D's of Roots

Sorting Pass

The purpose of the sorting pass is to manipulate the three lists so that they are correct with respect to the current value of the magnitude threshold N . In this pass, elements of the LIP may be moved to the LSP. Elements of the LIS are decomposed as necessary: a set in the LIS may be broken into type A or type B subsets and its roots may be moved into the LIP or LSP as appropriate. Each entry of the LIP is tested for significance with respect to n . If significant, a 1 is transmitted, a sign bit representing sign of that pixel is transmitted and pixel coordinates are moved to LSP. If not, then 0 is transmitted.

1. Process LIP.

a) For each coefficient (i,j) in LIP, $S_n(i,j)$ is output where $S_n(i,j) = 1$ when $\max |c(i,j)| \geq 2n$ or $S_n(i,j) = 0$ for other.

b) If $S_n(i,j) = 1$, sign of coefficient (i,j) : 0/1 is output and (i,j) is moved to the LSP.

2. Process LIS.

a) For each entry (i,j) in LIS and if the entry is of type D then output $S_n(D(i,j))$.

i) If $S_n(D(i,j)) = 1$ then for each $(k,l) \in O(i,j)$ output $S_n(k,l)$.

ii) If $S_n(k,l) = 1$, then add (k,l) to the LSP and output sign of coefficient: 0/1.

iii) If $S_n(k,l) = 0$, then add (k,l) to the end of the LIP.

b) If type L then output $S_n(L(i,j))$.

i) If $S_n(L(i,j)) = 1$ then add each $(k,l) \in O(i,j)$ to the end of the LIS as an entry of type D and remove (i,j) from the LIS.

Refinement pass

The n th MSB of the magnitude of each entry of the LSP, except those added in current sorting pass is transmitted. Note that at the end of 1st sorting pass no bits would be transmitted as a part of the refinement pass because the LSP contained no pixels prior to current sorting pass. Refinement pass is to be added keeping in mind the decoding or reconstruction. The refinement pass follows the sorting pass and outputs the bit corresponding to the current magnitude threshold for each of pixels in the LSP, which were not added in the immediately previous sorting pass.

1. Process LSP.

2. For each element (i,j) in LSP except those just added above in the sorting pass the n th most significant bit of coefficient is output.

Quantization Step Update

The quantization-step updates simply decrements N. That is, the magnitude threshold is decreased. The algorithm then returns to the sorting pass and continues. We can halt the algorithm at any time we wish, such as if the compressed data stream has reached the size we desire..N is decremented by 1 and the procedure is repeated from step 2 onwards.

1. Decrement n by 1.
2. Then go back to the Significance Map Encoding Step (Sorting Pass).

Decoding

An additional task done by decoder is to update the reconstructed image. For the value of n when a coordinate is moved to the LSP, it is known that

$$2^n \leq |C_{i,j}| < 2^{n+1}$$

So, the decoder uses that information, plus the sign bit that is input just after the insertion in the LSP, to set

$$C_{i,j} = \pm 1.5 * 2^n$$

Similarly, during the refinement pass the decoder adds or subtracts 2^{n-1} to $\hat{C}_{i,j}$ when it inputs the bits of the binary representation of $|C_{i,j}|$.

2.2. 2. Modified SPIHT Algorithm

The redundancy of original SPIHT is removed. The frequency coefficients obtained after transform are taken as input and the Modified SPIHT eliminate the correlation in same level sub bands[1].

Compression

1. Apply DWT on the original fingerprint image.
2. Make all the coefficients of this DWT positive.
3. Apply SPIHT on the DWT obtained after step 2 ignoring the sign bit and compressed bit stream is obtained.

Decompression

1. Apply inverse SPIHT on the bit stream recovered after step 2, again ignoring the sign bit since it has not been encoded.
2. Change the sign of all those pixels whose sign had been reversed during compression.

3. Apply IDWT on this transform to get back the original image.

2.2.3. Modified SPIHT with Run Length Encoding

RLE (Run Length Encoding) has to be applied to the resulting bit stream after SPIHT is applied to the DWT. It will help to reduce the size of the bit stream already generated. At the receiving end, the original bit stream can be easily recovered by applying inverse RLE.

3. Objective Quality Measures

Let $x(m, n)$ denotes the samples of original image, and $x'(m, n)$ denotes the samples of compressed image. M and N are number of pixels in row and column directions respectively [3].

Mean Square Error is given by:

$$\text{MSE} = \sum_{m=1}^M \sum_{n=1}^N (x(m, n) - x'(m, n))^2 / MN$$

Peak Signal To Noise Ratio is given as:

$$\text{PSNR} = 10 \log_{10} (255)^2 / \text{MSE}$$

4. Experimental Results

For evaluating the performance of the algorithms, we used an experimental database which consists of 50 images of 512 x 512 in bmp format. Results are shown on one of the fingerprint image using matlab platform, coded with Basic SPIHT and Modified SPIHT. The following result images are created in the 'work' folder of MATLAB.

afterreading: This is the image which has been read by the algorithm.

afterdwt: This is the image of the DWT of the original image.

afterspiht: This is the image of the resulting bit pattern after SPIHT has been applied on the DWT of the original image.

afterrle: This is the image of the compressed bit stream after applying RLE on the SPIHT output.

afterirle: This is the image of the DWT original bit stream reconstructed after applying inverse RLE on the received bit stream.

afterispiht: This is the image of the DWT, recovered after applying inverse SPIHT on the recovered bit stream.

afteridwt: This is recovered copy of the original image.



afterreading



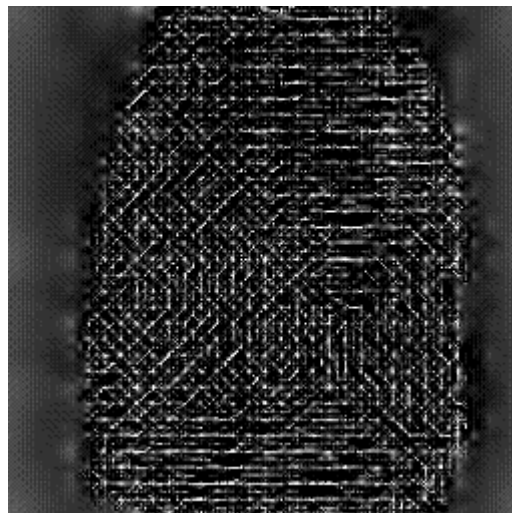
afterdwt



afterspiht



afterispiht



afteridwt

Figure 4.1: Basic SPIHT (1 bpp)

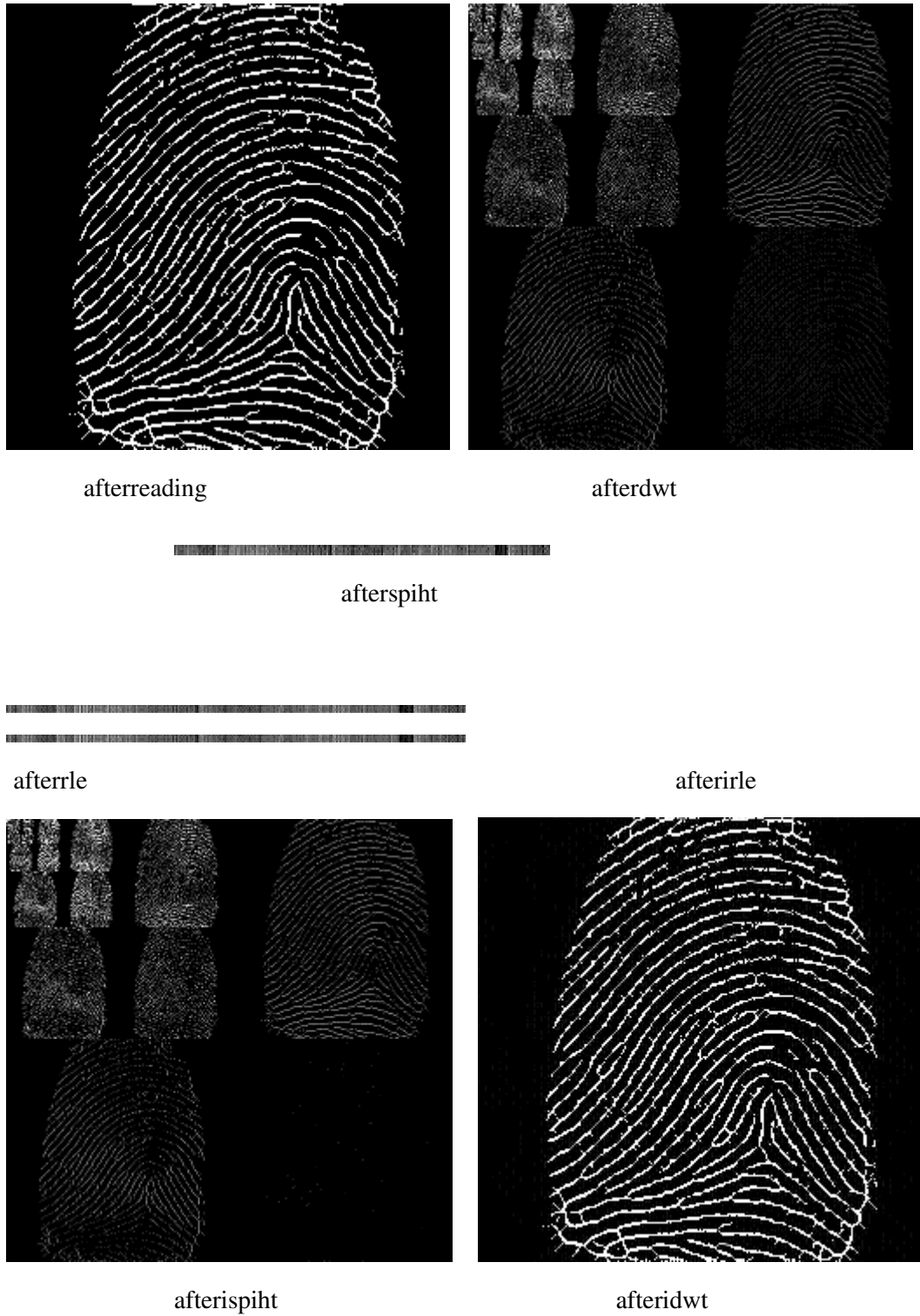
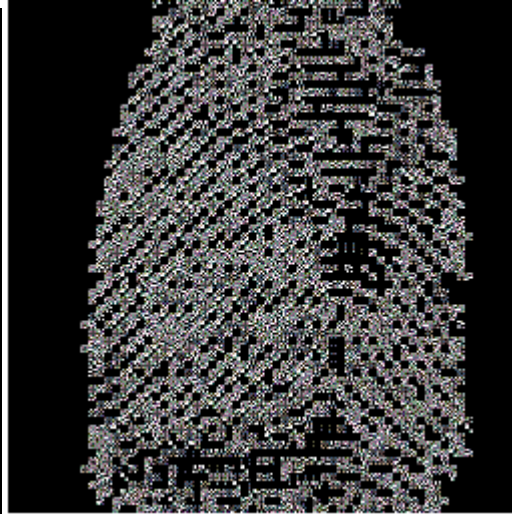


Figure 4.2: Modified SPIHT with RLE (1bpp)



Original Image



DCT



Compressed Output



Decompressed Output

Figure 4.3:JPEG

Table I shows the performance results for fingerprint test image using the algorithms. Further, it is observed that Modified SPIHT with RLE provide better results than Basic SPIHT and JPEG for the fingerprint image at various bit rate

i1.bmp basic	0.2	0.4	0.5	0.8	1
MSE	210.6489	205.8004	204.9470	202.0391	201.4177
PSNR	24.8952	24.9963	25.0144	25.0765	25.0898
i1.bmp modified	0.2	0.4	0.5	0.8	1
MSE	89.4559	74.3574	71.0722	70.0243	69.0232
PSNR	28.6147	29.4176	29.6138	29.7085	30.004
i1.bmp jpeg	0.2	0.4	0.5	0.8	1
MSE	577.0869	440.9861	387.3844	267.5766	239.9550
PSNR	20.5184	21.6866	22.2494	23.8563	24.3295

Table I: Performance Results

5. CONCLUSION

The comparative study shows that DCT based JPEG has higher processing speed than other two algorithms. But at higher compression ratio it shows blocking artifacts that affect the performance of automated systems that attempt to trace out ridges to help identify the fingerprint. Blocking artifacts are absent in reconstructed images by wavelet based Basic SPIHT and Modified SPIHT methods.

From the experimental results, it is seen that Modified SPIHT with Run Length Encoding gives better PSNR as compared to Basic SPIHT Algorithm and JPEG. Comparison of Modified SPIHT and Basic SPIHT using PSNR and MSE as image quality measure shows that Modified SPIHT achieves higher picture quality than Basic SPIHT and JPEG for all compression ratios and test images.

REFERENCES

- [1] Anilkumar V. Nandi, Dr.R.M.Banakar, "Hardware modeling and implementation of Industrial and Information systems,2007,ICIIS 2007, pp.329-334.
- [2] R. Sudhakar, R. Karthiga, S. Jayaraman, "Fingerprint Compression Using Contourlet Transform with Modified SPIHT Algorithm". Journal of Electrical and Computer Engineering, Vol. 5, No. 1, Dec 2006, pp 74-84.
- [3] Bhawna1 Rani, R K Bansal , Dr Savina Bansal," Comparison of JPEG and SPIHT Image Compression Algorithms using Objective Quality Measures", Multimedia, Signal Processing and Communication Technologies, 2009. IMPACT '09. International Publication, pp.90-93.
- [4] Brahim, T. Melit, A. KHelifi, "Improvements to SPIHT for Lossless Image Coding", 3rd International Conference on Information and communication Technologies, 2006, ICTTA 2006, pp 90-93.

- [5] A. Said, W. A. Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 6, No. 3, pp. 243-249.
- [6] Tareq Aziz, M.A. Illaqqe "Comparative study of different fingerprint compression schemes", IEEE 2005, pp 562-565.
- [7] Anil Jain, Arun Ross, Salil Prabhakar, "Fingerprint matching using minutiae and texture features," Int'l conference on Image Processing (ICIP), pp. 282-285, Oct. 2001.
- [8] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity authentication system using fingerprints," Proc. IEEE, Vol. 85, pp. 1365-1388, Sept. 1997.
- [9] Arun Ross, Anil Jain, James Reisman, "A hybrid fingerprint matcher," Int'l conference on Pattern Recognition (ICPR), Aug 2002.
- [10] Arun Ross, Sarat Dass, Anil Jain, "A deformable model for fingerprint matching," Pattern Recognition Vol. 38, pp. 95-103, 2005.
- [11] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based Fingerprint Matching," IEEE Trans. Image Processing, Vol. 9, No. 5, pp. 846-859, May 2000.
- [12] A. K. Jain, L. Hong, Y. Kulkarni "A Multimodal Biometric System using Fingerprint, Face, and Speech," Proc. 2nd Int'l Conference on Audio- and Video-based Biometric Person Authentication, Washington D.C., pp. 182-187, 1999.