Enhancement of Denoising Methodology by Embedding MS-VST with EZW

B. Doss
Electronics & Communication Engineering
Jawaharlal Nehru Technological University
Ananthapur, India

M. Praveen Kumar
Electronics & Communication Engineering
Jawaharlal Nehru Technological University
Ananthapur, India

***Abstract* – The variance stabilizing transform (VST) is a frequently used denoising algorithm which is applicable in many areas like medical, defence and commercial applications. As the image data transmission issues are raising day by day we are adopting so many new methods as the combinations of VST algorithm. The embedded zero tree wavelet algorithm (EZW) is basically used for compression techniques from the past years. As its accuracy level is high we are going to embed this technique to current VST algorithm. Therefore, it is suitable for denoising applications. Combining VST with the EZW leads to good Poisson image denoising algorithms, even for low intensity images. We present a simple approach to achieve this and demonstrate some simulation results. The results show that the VST combined with the EZW is an optimistic technique for better Poisson denoising.**

Keywords – Variance stabilizing transform, EZW, wavelet transform, zero tree, poisson denoising.

I. INTRODUCTION

Poisson noise images arise from a variety of applications involving astronomy & astrophysics where detection of photon particles is performed [1]. The VST is an optimum technique for Poisson denoising in images. The VST fixes the variance of the poisson image to constant, ensue a homoskedastic gaussian. After converting, quality denoising methods are employed before VST to reestimate the final image. The execution of VST deteriorates for low intensity images. To improve the performance of VST the signal to noise ratio should be increased before stabilization. This can be achieved by prefiltering process where the noise is averaged out by low pass filters where input image is transmitted with multi scale in multi directions leading to a transform called MS-VST [3]. MS-VST's were implemented on wavelets, Ridgelets, first generation curvelet transforms and moreover on second generation curvelets transforms called FDCT [4]. It is observed that curvelets put up the best performance for poisson noise removal. Moreover it is observed that MS-VST based poisson denoising with non subsampled contourlet transform puts up good performance for noise removal even for low intensity images. Its synopsis is quite comparable with the curvelets but optimises at the rate of high margin.

In this paper we propose a method which combines MS-VST with EZW (Embedded Zero Wavelets) implemented with an Embedded zero tree coding algorithms. The variance of the image is calculated at the zero positions of the mean value. The embedded zero tree wavelet algorithms are an effective image compression algorithm with the property that the bits in the bit stream are generated in order of importance providing fully embedded code. The EZW combined with MS-VST has less redundancy and even computational complexity than the curvelets transforms. We apply our proposed method EZW based MS-VST to denoise poisson noise images and shown some results. The results proposed in this paper shows that EZW based MS-VST is veritably a favourable algorithm for poisson noise removal.

This paper is organised as follows. Section II describes briefly about MS-VST. In section III we discuss briefly about wavelets and EZW coding structure and its algorithm. The proposed method EZW based MS-VST algorithm is discussed in section IV. The procedure involved in poisson noise removal with proposed method is discussed in section V. Section VI and VII deals with simulation results and conclusion.

II. MULTISCALE VARIANCE STABILIZING TRANSFORM

A poisson process $x := (X_i)_{i \in Z^d}$ is defined as $X_i = \alpha U_i + V_i$, $U_i \sim p(\lambda_i)$, $V_i \sim \mathcal{N}(\mu, \sigma^2)$ where $\alpha > 0$ is the overall gain of the detector, U_i & V_i are assumed to be mutually independent. The discrete filtered process for a given filter coefficient h can be given as $Y_i := \sum_j h[j] X_{i-j}$ where X & Y are used one among from X_i & Y_i respectively. We still define by τ_k and can be given as $\sum_i (h[i])^k$ for k = 1, 2, 3……… and so on. We consider image to be locally homogeneous i.e. $\lambda_{j-i} = \lambda$ for all i within the support of h. It is observed that variance of Y (Var [Y]) is directly proportional to the intensity λ. To stabilize variance we adopt a transformation $Z := T(Y)$ such that Var [Z] is constant irrespective of the value λ.

The VST for a filtered image defined in [3] can be given as

$$T(Y) := b.sgn(Y + c)\,|Y + c|^{1/2}, b \neq 0, c \in \mathbb{R} \quad (1)$$

where b is the normalizing factor. Lemma I confirms our heuristics that T is indeed a VST filtered poisson process with a nonzero mean filter, *T(Y)* is asymptotically normally distributed with a stabilized variance λ.

Lemma I (square root as VST): If $\tau_1 \neq 0, \|h\|_2, \|h\|_3 < \infty$, then we have

$$sgn(Y + c)\sqrt{|Y + c|} - sgn(\tau_1)\sqrt{|\tau_1|\lambda} \xrightarrow[\lambda \to +\infty]{\mathcal{D}} \mathcal{N}\left(0, \frac{\tau_2}{4\,|\tau_1|}\right) \quad (2)$$

with sgn(•) the signum function.

This result holds true for any $c \in \mathbb{R}$, that controls the convergence rate. From [4] by defining constants b & c the filtered poisson process can be given as

$$T(Y) - b.sgn(\tau_1)\sqrt{|\tau_1|\lambda} \xrightarrow[\lambda \to +\infty]{\mathcal{D}} \mathcal{N}(0,1) \qquad (3)$$

and the rate of convergence of the variance is $O(\lambda^{-2})$ by fixing c.

In summary MS-VST consists of the following steps

1) **Transformation:** Poisson image is transmitted through MS-VST which transforms into Gaussian distribution with constant variance.
2) **Detection:** significant details are detected by hypothesis test.
3) **Estimation:** Determine the reconstructed image using ℓ_1 optimization technique using hybrid iterative steepest descent algorithm.

III. EMBEDDED ZERO WAVELETS

The embedded zero tree wavelet algorithm (EZW) is a simple, effective compression algorithm bearing a property that the bits in the bit stream are generated in order of importance, achieving a fully embedded code [8]. The embedded code represents a sequence of binary decisions that distinguish an image from the "null" image. Using an embedded coding algorithm, an encoder can terminate the encoding at any point thereby a target rate or target distortion metric to be met exactly. Also, given a bit stream, the decoder can cease decoding at any point in the bit stream and still produce exactly the same image that would have been encoded at the bit rate corresponding to the truncated bit stream. In addition to produce a fully embedded bit stream, EZW consistently produces compression results that are competitive with virtually all known compression algorithms on standard test images.

The EZW algorithm [10] is based on four key concepts:

1) A discrete wavelet transform or hierarchical sub-band Decomposition,
2) Prediction of the absence of significant information across scales by exploiting the self-similarity inherent in images,
3) Entropy - coded successive - approximation quantization,
4) Universal lossless data compression which is achieved via adaptive arithmetic coding.

The EZW encoder is a special type of encoder used to encode signals of any dimensions. It is used to encode the signals such as image, sound etc. even, it couldn`t beat out the popular type of encoders in the market. It can have its own applications in the embedded encoded systems. Hence it could maintain the illegal access of the data from the systems.

The EZW encoder provides the advantages of Denoising; lower BW, simply implemented algorithms, which are well explained in the following sections. There it can provide a lot of room for the embedded type of encoding.

A. Wavelet Transform: The wavelet transform has generated a tremendous interest in both the oretical and applied areas, especially over the past years. Wavelet theory involves representing general functions in terms of simpler, fixed building blocks at different scales and positions. It provides useful synthesis for subband filtering techniques, quadrature mirror filters; pyramid schemes e.t.c. let us consider an image of size $M \times N$ i.e. to be decomposed into four subbands after passing through low pass filter and high pass filter. The four subbands are termed as LL, LH, HL & HH. The subbands obtained by low pass filtering is referred to as LL contains horizontal details of the image and by high pass filtering referred to as HH contains vertical details of an image.

The basic wavelet function termed as $h(x)$ [6] is given as

$$h_{m,n}(x) = a_0^{-m/2}\, h(a_0^{-m}x - nb_0) \qquad (4)$$

The discrete wavelet transform for an arbitrary function $f(x) = L^2(R)$ can be given as

$$W_{m,n} = \int_{-\infty}^{\infty} f(x) h_{m,n}(x) dx \qquad (5)$$

After performing wavelet transform the spatial distribution of the wavelet coefficients which describes the original image with different angles is shown in Fig. (1) and frequency distribution for second wavelet transform for single subband of Fig. (1) is shown in Fig. (2).

LL1	LH1
HL1	HH1

Fig. (1) Frequency distribution after first wavelet transform

<table>
<tr><td>LL2</td><td>LH2</td><td rowspan="2">LH1</td></tr>
<tr><td>HL2</td><td>HH2</td></tr>
<tr><td colspan="2">HL1</td><td>HH1</td></tr>
</table>

Fig. (2) Frequency distribution after second wavelet transform

The wavelet coefficients across the subbands bear a definite spatial relationship, utilized for space frequency localization and encoding of wavelet coefficients.

B. EZW Encoding:

Embedded zero wavelet encoding scheme is implemented to work with wavelet transforms. It is specifically designed to operate on 2D signals (images) but even it is used for other dimensional signals. Principle of EZW encoding is based on progressive encoding also referred to as embedded encoding which compresses the image into sequence of bit streams as the accuracy is increasing, which reveals that each time bit is added, the decoded image will contains more detail information. EZW coding and with some optimization techniques, results in an effective image compressor with the property that, compressed data stream can have any bit rate desired. Any bit rate is possible when it has loss of information, even then lossless compression is possible with an EZW encoder but with peculiar results.

C. Relation between subbands:

Every coefficient in an echelon subsystem at a given scale is possible to relate a set of coefficients to the next finer scale of identical coefficients, except the higher frequency subbands. Since beyond this there is no existence of finer scale subbands [7].

Descendants: The coefficients at coarser scale are called parent and the set of all these coefficients at all the finer scales with identical orientations and spatial locations is termed as descendants.

Ancestors: The coefficients corresponding to the coarser scale to the same spatial location at the next finer scale are called children and the set of all these coefficients at all the finer scales with identical orientations and spatial locations is termed as ancestors.

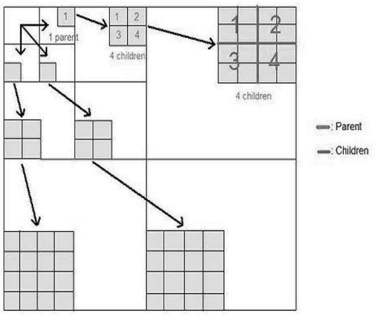

Fig (3) Parent children dependencies of subbands

EZW encoding improves the basic wavelet encoding by efficiently representing portions of encoded symbol stream that can be safely eliminated because their aggregate energy falls below a predetermined threshold. A wavelet coefficient *x* is said to be insignificant with respect to a given threshold *T*, if |*x*| < *T*. The zero tree is based on the hypothesis that if a wavelet coefficient at a coarse scale is insignificant with respect to a threshold, then all wavelet coefficients of the same orientation in the same spatial location at the finer scale are likely to be insignificant with respect to the same threshold. More specifically, in a hierarchical subband system, with the exception of the highest frequency subbands, every coefficient at a given scale can be related to a set of coefficients at the next finer scale of similar orientation. The data structure of the zero tree can be visualized in Figure (4). Given a threshold *T* to determine whether or not a coefficient is significant, a coefficient *x* is said to be an element of a zero tree for the threshold *T* if itself and all of its descendents are insignificant with respect to the threshold *T*.

Hence for a given a threshold, any wavelet coefficient could be represented in one of the four data types: zero tree root (ZRT), isolated zero (IZ) (it is insignificant but its descendant is not), positive significant (POS) and negative significant (NEG).

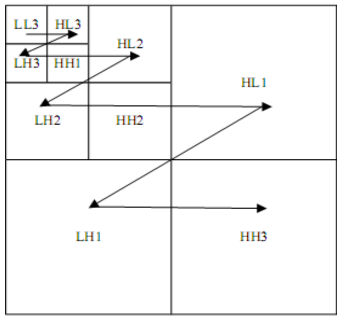

Fig (4) Scanning order embedded zero tree wavelet algorithms [6]

D. Flow chart for encoding map:

The encoding of the coefficients into one of the above four symbols is illustrated in the Figure (5)

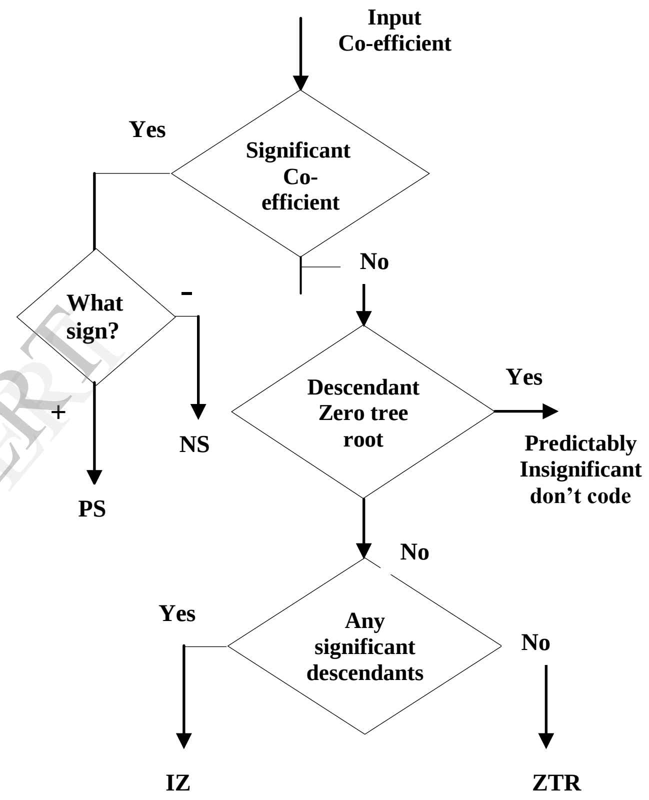

Fig (5) Flow chart for encoding map

E. The Zero Tree:

The EZW encoder [8] is based on two important observations:

1. When an image is wavelet transformed the energy in the subbands decreases as the scale decreases, hence the wavelet coefficients are average and smaller in the higher subbands than the lower subbands.
2. Large wavelet coefficients are more important than the smaller wavelet coefficients.

These two observations are exploited by the EZW encoding scheme by coding the coefficients in decreasing order in several passes. A threshold is choosen for every pass. If a wavelet coefficient is above than the threshold it is encoded and removed from the image, if it is smaller left for next pass. When all the wavelet coefficients are analyzed completely, threshold is lowered and the image is scanned again. This method is applicable to use the dependency between the wavelet coefficients across different scales to encode large parts of the image which are below threshold. This process enters now in to zero tree.

F. EZW Algorithm:

Step – 1: Determine the initial threshold

$$t_i = 2^{\lfloor \log_2 (MAX(|\gamma(x,y)|)) \rfloor} \quad (6)$$

Step – 2: a coding loop is maintained

Threshold = initial_threshold
do {
dominant pass (image);
subordinate pass (image);
threshold = threshold / 2;
} while (threshold > minimum_threshold)

Step – 3: Dominant pass (significance pass)

- Wavelet coefficients on the dominant list are compared with initial threshold to determine significance and if significant, sign. The resulting significance map is zero tree coded and sent.

Step – 4: code using any four symbols

1. Zero tree root
2. Isolated zero
3. Positive significant
4. Negative significant

Step – 5: Entropy code using adaptive AC,

- Return each coefficient, if it is coded as significant (pos or neg)

1. Put its magnitude on the subordinate list.
2. Remove it from the dominant list.

Step – 6: Subordinate pass (Refinement pass)

- Provide one more bit on the magnitude on the subordinate list as follows.
  1. Halve the quantizer cells.
  2. If magnitude is in upper half of old cell provide “1”.
  3. If magnitude is in lower half of old cell, provide “0”.

Step – 7: Entropy code sequence

- Return 1’s and 0’s using adaptive AC, and send stop, when bit budget is exhausted.

Encoded stream has embedded in it all lower rate encoded versions. Thus, encoding / decoding can be terminated prior to reaching full rate version.

IV. MS-VST WITH EZW CODING

The proposed method of embedding MS-VST with the EZW coding is shown in Fig. (3). In this method as the performance of the VST deteriorates for low intensity images, prefiltering process is performed before stabilizing the variance. Filtering process in achieved by means of low pass filters using 2D filter banks having M channels. The pass band for the m^{th} filter bank is $\left[-\pi/2^m, \pi/2^m\right]^2$. Upon 2D DFT is applied to each low pass band and subsequently EZW encoding and decoding is performed on each filtered image. Taking the IDFT results in an each coefficient of the EZW encoded and decoded coefficient. The low pass filters are choosen over the usual laplacian iterative pyramid to avoid cumulative errors. The EZW coding block performs both encoding and decoding process and outputs MS-VST + EZW coefficients.

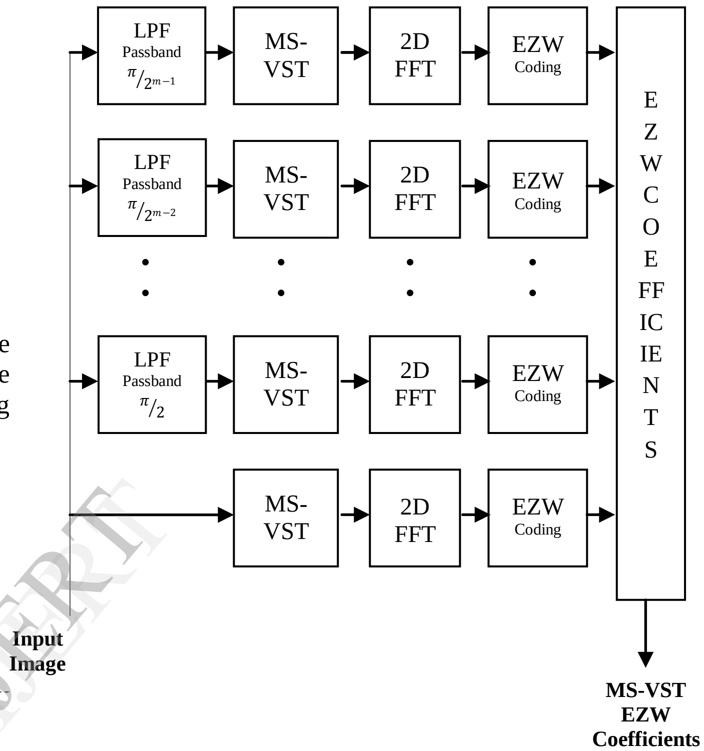

Fig (6) MS-VST with EZW coding Structure

Low pass filter design: The additional filters [8] used will cause errors in EZW coding mechanism and may destroy its reconstruction property. This is not a serious problem because in case of denoising method, direst reconstruction is not possible. The reconstruction process is performed indirectly by means of $\ell_1$optimization techniques. As explained in Section V and are used only for multiresolution support.

The resulting EZW coding mechanism for poisson noise denoising for an image X is given by

$$\eta(\omega) = \begin{cases} \omega + \tau_h - \frac{\tau_h}{2k+1}, \omega < -\tau_h \\ \frac{1}{(2K+1)t^{2K}} \omega^{2k+1}, |\omega| \le \tau_h \\ \omega - \tau_h + \frac{\tau_h}{2K+1}, \omega > \tau_h \end{cases} \quad (7)$$

where ω expresses the original wavelet coefficient and $\eta(\omega)$ represents the wavelet coefficient after thresholding process.

We have computed the variance of the EZW coding and decoding coefficients obtained by the proposed scheme. We have used Fisher and Monte Carlo simulations by giving the poisson noise as inputs to the filter bank. This process is repeated for different low intensity images and the results are shown in Fig (7). More over filters with narrower Passband, the SNR is improved more and hence they converge faster.

V MS-VST USING EZW DENOISING

EZW compression/decompression is very similar to wavelet shrinkage. Wavelet shrinkage is a way of noise reduction by trying to remove the wavelet coefficients that correspond to noise. Since noise is uncorrelated and usually small with respect to the signal of interest, the wavelet coefficients that result from it will be uncorrelated too and probably be small as well. The idea is therefore to remove the small coefficients before reconstruction and so remove the noise. Of course this method is not perfect because some parts of the signal of interest will also result in small coefficients, indistinguishable from the noisy coefficients that are thrown away. But if the selection criteria are well chosen, based on statistical properties of the input signal and the required reconstruction quality, the result is quite remarkable.

There are two popular kinds of wavelet shrinkage: one using hard thresholding and one using soft thresholding. For hard thresholding you simply throw away all wavelet coefficients smaller than a certain threshold. For soft thresholding you subtract a constant value from all wavelet coefficients and you throw away everything smaller than zero (if we consider only the positive values).

In the following, when we use EZW, we mean the reorganisation of the wavelet coefficients as described above only, not the compression part. The compression doesn't change anything; it just makes it less clear what happens. When we do lossy EZW, we throw away the coefficients that you don't need to meet the quality criteria you set for the reconstruction. These coefficients are the small ones, because these are the ones that add the extra detail. Due to the way EZW works, these are also the coefficients that are encoded at the end of the EZW stream. The threshold for lossy EZW is often based on visual criteria and acceptable PSNR after reconstruction.

Since we simply throw away the small coefficients, lossy EZW looks a lot like hard thresholding, but it isn't, because the small coefficients at the end of the stream are used also to lift the already decoded coefficients to their original level. So if we do not have these small coefficients, all the other coefficients will be too small. Furthermore, this difference is the same for all coefficients. This means that lossy EZW is identical to soft thresholding.

We used PSNR and NMISE to measure the performance of different MS-VST algorithms and compared these results in Table-I. For an image of size $M \times N$ parameters can be computed from the below equations as

$$\text{NMISE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(f(i,j)-\bar{f}(i,j)\right)^2 / f(i.j) \quad (8)$$

$$\text{PSNR} = 20\ \log\frac{512^2}{MSE}\ (dB) \quad (9)$$

where $\bar{f}(i,j)$ is the gray level profile of the pixel for reconstructed image and $f(i.j)$ is the gray level profile of the pixel for original image.

TABLE – I

Comparisons of PSNR for different MS-VST algorithms

S. no	Algorithm	PSNR (dB's) (*Cameraman*)	PSNR (dB's) (*Living room*)
1	MS-VST	55.86	54.44
2	MS-VST+SNSCT	45.50	50.13
3	MS-VST+NSCT	35.65	36.48
4	MS-VST + CVT	24.73	24.72
5	MS-VST + FDCT	45.50	44.73
6	MS-VST + EZW	63.24	60.10

TABLE – II

Comparisons of NMISE for different MS-VST algorithms

S. no	Algorithm	NMISE (*Cameraman*)	NMISE (*Living room*)
1	MS-VST	0.35	0.31
2	MS-VST+SNSCT	0.24	0.29
3	MS-VST+NSCT	0.20	0.27
4	MS-VST + CVT	0.37	0.33
5	MS-VST + FDCT	0.25	0.18
6	MS-VST + EZW	0.16	0.14

TABLE-I compares the PSNR value with different MS-VST based methods, it is observed our proposed method shows the maximum Signal to ratio compared the second generation curvelet transforms; in addition it has least NMISE value shown in TABLE-II. We also demonstrated the amount of added noise and noise removed based on percentage levels on the images for different MS-VST based methods listed in TABLE – III.

(a)

(b)

(c)

(d)


(e)


(f)


(g)


(h)

Fig. 8. Denoising of poisson noise from "Cameraman" image (image size: *512 × 512*). (a) Original image; (b) Poisson count image (Noise added in terms of percentage 34%); (c) Anscombe based MS-VST denoised image (Noise out = 25%; NMISE = 0.35); (d) MS-VST + Semi NSCT denoised image (Noise out = 24%; NMISE = 0.24); (e) Image denoised by MS-VST + NSCT (Noise out = 21%; NMISE = 0.20); (f) Curvelets based MS-VST denoised image (Noise out = 19%; NMISE = 0.37); (g) second generation curvelet FDCT based MS-VST denoised image (Noise out = 18%; NMISE = 0.25); (h) Wavelet based compression technique for denoising Poisson noise based on MS-VST + EZW denoised image (Noise out = 16%; NMISE = 0.16).

TABLE – III

% Noise added and removed on original images

S. no	Algorithm	Noise in	Noise out
1	MS-VST	34	25
2	MS-VST+SNSCT	34	24
3	MS-VST+NSCT	34	21
4	MS-VST + CVT	34	19
5	MS-VST + FDCT	34	18
6	MS-VST + EZW	34	16

In addition we are demonstrating our results in graphical view, where we are comparing the noise level with PSNR for different state of arts of MS-VST algorithms.

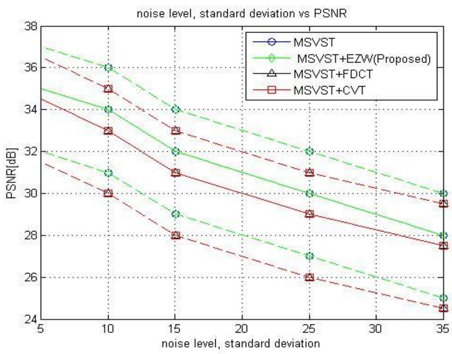

Figure (7) Variations of PSNR for MS-VST based methods

VI SIMULATION RESULTS

The experimental results for our proposed method MS-VST+EZW for poisson noisy images are demonstrated. In achieving our proposed method we used MATLAB simulink tool. We use the images of size *512 × 512* for both the images "*Cameraman"* and *"Living room"*. All the experiments are performed by encoding and decoding an actual bit stream to verify the correctness of the algorithm. The model is initialized at each new threshold for each of the dominant pass and subordinate pass. Results are plotted in TABLE – I with respect to PSNR and in TABLE – II is with NMISE (Normalized integrated square error). We also demonstrated the percentage of noise added and removed during the filtering process in TABLE – III. The EZW encoder was applied on the standard Camera man image and Living room image. Quotes of PSNR for "Living room" image are so abundant throughout the image coding literature that it is difficult to definitively compare the results with other transforms.

From TABLE – III it is evident that for the noise added for various transforms and to our proposed method is 34; it reveals that our proposed method is effective method for detecting noise components in the image which is only 16. Even it is proven with PSNR. For the Living room image the performance using EZW encoder is substantially better than the other images. Figure (8) shows the results corresponding to the image *"Cameraman"*. A poisson noise applied image is shown


(a)


(b)


(c)


(d)

(e) (f) (g) (h)

Fig. 9. Denoising of poisson noise from “Living room” image (image size: 512×512). (a) Original image; (b) Poisson count image (Noise added in terms of percentage 34%); (c) Anscombe based MS-VST denoised image (Noise out = 25%; NMISE = 0.31); (d) MS-VST + Semi NSCT denoised image (Noise out = 24%; NMISE = 0.29); (e) Image denoised by MS-VST + NSCT (Noise out = 21%; NMISE = 0.27); (f) Curvelets based MS-VST denoised image (Noise out = 19%; NMISE = 0.33); (g) second generation curvelet FDCT based MS-VST denoised image (Noise out = 18%; NMISE = 0.18); (h) Wavelet based compression technique for denoising Poisson noise based on MS-VST + EZW denoised image (Noise out = 16%; NMISE = 0.14).

in Fig. (8b). The MS-VST based transform in shown in Fig. (8c), MS-VST + SNSCT in Fig. (8d), the estimation of MS-VST+NSCT [11] is shown in Fig. (8e), the estimation with respect to curvelets is shown in Fig. (8f), with respect to second generation curvelet transforms in Fig. (8g) and encoding and decoding with respect to EZW in Fig. (8h). The MS-VST + SNSCT yield better performance by bringing out the high frequency details in the image with NMISE value being smaller. Similarly Fig. (9) Shows the results obtained for *“Living room”* image. For this method also encoding and decoding process has become difficult to compare with the other transforms. The results plotted for this image also quite comparable with the camera man image with slight variation. The NMISE value for both the images in our proposed method is quite achievable value to say that a better denoising is achieved. With these results we can say that our proposed method MS-VST + EZW is a promising algorithm for better denoising of poisson noise form images.

VII CONCLUSIONS

We have presented an approach to combine the variance stabilizing transform (VST) with the Embedded Zero Wavelets compression technique to achieve denoising of poisson noise images. This method uses the efficient algorithm for poisson denoise process using compression technique. This method of approach araised from basic wavelet shrinkage which leads to noise reduction even data compression. Results of our proposed method MS-VST based EZW tree coding technique proves that this combined algorithm is a promising one to have better denoising of poisson noise from images.

REFERENCES

[1]. J.-L. Starck, F. Murtagh, and A. Bijaomi, *Image Processing and Data Analysis: The Multiscale Approach.* Cambridge Univ. Press, 1998.
[2]. F.J. Anscombe, “The transformation of poisson, binomial and negative binomial data,” *Biometrika,* vol. 35, pp. 246-254, 1948.
[3]. B. Zhang, J.-L. Starck, “Wavelets, ridglets, and curvelets for poisson noise removal,” *IEEE Trans. Image Process.,* vol. 17, pp. 1093-1108, July 2008.
[4]. E. Candes, L. Demanet, D. Donoho, and L. Ying, “Fast Discrete curvelet transforms,” *SIAM Multiscale Modelling and Simulation,* vol. 56, no. 3, pp. 861-899, 2006.
[5]. S. Pallakal and K. M. M Prabhu, “Poisson noise removal from images using Fast Discrete Curvelet Transform,” *IEEE Image process,* 2011
[6]. Genshum Wan, Xuehua Song, Riccardo Bettati “An improved EZW algorithm and its applications in Intelligent Transport Systems,” 2014
[7]. IIT Kharagpur, Lesson 13 “A Zero Tree Approach”, version – 2.
[8]. C. Valens, “Embedded Zero Tree Wavelet Encoding,” 1999
[9]. J. Shapiro, “Embedded image coding using zero trees of wavelet coefficients,” *IEEE Trans. Signal processing,* vol. 41, pp. 3445-3462, Dec. 1993.
[10]. Pardeep singh, Sugandha Sharma, Bhupinder Singh, “Comparative analysis of embedded zero trees and fractal image compression techniques,” vol. 2. Issue. 2. Feb. 2012.
[11]. A. L. da Cunha, J. Zhou, and M. N. Do, “The Nonsubsampled contourlet transform, Theory, design and applications,” *IEEE Trans. on image Process.* vol. 15, pp. 3089-3101, Oct. 2006.