Mannava Srinivasa Rao, Boppana Swati Lakshmi, Dr.Panakala Rajesh Kumar/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.1936-1941

Wireless Image Transmission over Noisy Channels Using Turbo Codes and De-noising Filters

Mannava Srinivasa Rao*, Boppana Swati Lakshmi, Dr.Panakala Rajesh Kumar*****
Department of ECE, PVP Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India

ABSTRACT

In today's world transmission of data i.e. voice and image plays vital role in day to day communication. While transmitting the voice or image over wireless channels there is higher probability of information getting corrupted because of various types of noises presence in the channel. This paper focusing on transmission of image through wireless channels. The impact of noise can be minimized by using effective channel coding technique called Turbo coding and de-noising filters i.e. wiener or median filter. These turbo codes are proved to be best to minimize the probability of bit errors & filtering approach is proved to be best to de-noise the image by obtaining higher PSNR. Simulated results show there is a clear improvement in quality of image using turbo codes with or without filters.

Keywords—Image compression, Bit plane Slicing, Turbo coding, De-noising Filters.

I. INTRODUCTION

To increase the reliability of data transmission over wireless channels, channel coding is required before transmission of data over noisy channel. More number of new coding techniques have been developed recently for efficient coding of images and video. Before channel coding image is converted in to binary format using Bitplane Slicing technique. In Bitplane Slicing the original image is partitioned in to 2^Nquantization levels, where N represents number bit planes. Then each of the bit plane is encoded by Turbo encoder [1] and transmitted over Additive White Gaussian noise (AWGN) channel. Compression of an image may lead to the degradation of quality of an image but saves the bandwidth. At the receiver side each of these noisy planes are evaluated using an iterative Turbo decoder and de-noising filters. These planes are re-assembled taking into consideration of neighborhood relationship between the pixels in the image. Bit plane Slicing [2] is an efficient method for compression of an image and iterative turbo decoding is proved to be best for decoding of an image, by considering the stochastic properties and Neighborhood relationship between the pixels. Turbo coding is used to provide better bit error rate performance and robust transmission of an image even if the channel is noisy.Turbo codes are mainly attractive for high-data-rate services due to the relatively long interleavers.

II. PROPOSED ARCHITECTURE

The main aim of this system is to transmit an image with effective coding as well as efficient reconstruction even in a noisy environment. The system consists of a Bitplane Slicer, Turbo encoder in the transmitter part, the receiver comprises of an iterative turbo decoder, De-noising filters, and image combiner section as shown in Fig.1.A 2D image will be applied as an input to the system
.

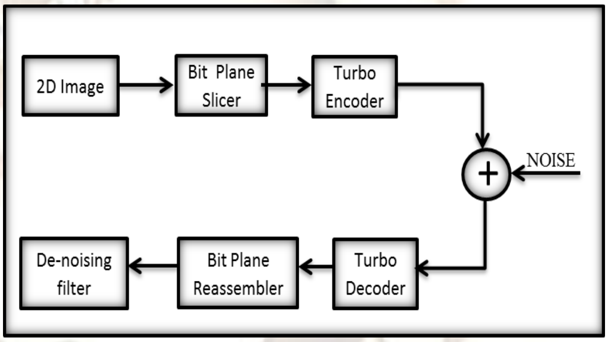


Fig.1: Proposed Architecture

For the applied image the binary correspondence of each pixel amplitude value is grouped in bit planes. Then these bit planes are encoded by using turbo coder and transmitted over a noisy channel. Using iterative turbo decoder and De-noising filter we can effectively reconstruct the image.

2.1 BIT PLANE SLICING/ RE-ASSEMBLING

Bitplane Slicing is used for compression [4] of data in the image processing area. Assume that the image is composed of N bit planes ranging from '0' to 'N-1'. Plane 0 is the Least Significant Bit (LSB) plane and plane N-1 is the Most Significant Bitplane. The slicing of an image shows that higher order bits contain visually significant data. So the plane N-1 corresponds exactly with an image threshold at gray level 2^{N-1}.Bit plane combining is the reverse process of the slicing. These planes are recombined in order to reconstruct the image. But it is not needed to take into consideration of all the slice contributions.

(a)

(b)

(c)

Fig. 2: The effect of various Slice Combinations
BP7 – Most Significant Bit plane
BP0 - Least Significant Bit plane

- **(a)** Image representing Bit plane 7 +Bit plane 6
- **(b)** Image representing Bit plane 7 + Bitplane 6 + Bitplane5
- **(c)** Image representing Bit plane 7+Bit plane 6+Bit plane 5 +Bitplane 4

For the importance of data rate, some planes can be ignored until the changes in gray level have unacceptable impact on the image. This approach will increase the data rate. Fig. 2 shows how the combinations of the slices contribute to recovery of the image. In this paper, the image is sliced to 4 planes i.e. each pixel in the image is represented by 4 bits (or 16 grey levels).

Imagine that the image is composed of four bit planes, ranging from plane 0 for the least significant bit (LSB), to plane 3 for the most significant bit (MSB). Note that the most Significant bit plane contains visually significant data. Transforming the image in a binary fashion is very suitable before transmission. If the image has been considered without being sliced, then the neighbourhood relationship would have been lost. So, it would be useless at the receiver side. When the image is sliced first and then coded and transmitted, the neighbourhood properties would be evaluated.

2.2 TURBO ENCODER

In figure 3 the turbo encoder [8]employs two systematic recursive convolutional encoders connected in parallel, with a random interleaver preceding the second recursive convolutional encoder. The information bits are encoded by both encoders. The first encoder operates on the input bits in their original order, while the second encoder operates on the input bits as permuted by the random interleaver. Depending on the code rate desired, the parity bits from the two constituent encoders are punctured before transmission. For example a turbo encoder of rate 1/3 means all parity bits are transmitted, whereas, for a rate 1/2 turbo code, the parity bits from the constituent codes are punctured alternately.

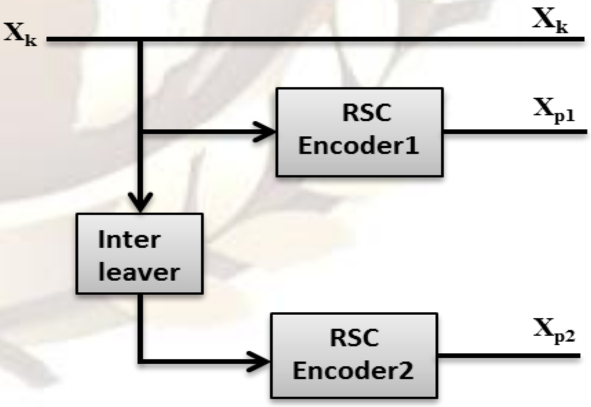


Fig. 3: Rate 1/3 Turbo Encoder

2.3 TURBO DECODER

In turbo decoder, [4]-[5] Soft information out of the demodulator regarding the systematic bits and parity bits from the first constituent encoder is sent to the first decoder. The first decoder generates soft-decision likelihood values for the information bits that are passed to the second decoder as *a priori* information after reordering in accordance with the turbo interleaver. In addition, the second decoder accepts the demodulator output regarding the systematic bits and the parity bits from the second constituent encoder. The second decoder improves on the soft-decision likelihood values for the information bits, which are then fed back to the first decoder to repeat the process. The process can be iterated as many times as desired. However, only a relatively small number of iterations are usually needed, since additional iterations generally produce diminishing returns. Hard decisions on the systematic information bits are made after the last decoder iteration is completed. Efficient algorithms, were developed for turbo decoding namely

1. Map decoding
2. Log map decoding
3. Maximum log map decoding

The performance of map decoding algorithm is superior with compared to log map and maximum log map decoding algorithm. Therefore the BCJR (Bahl-Cocke-Jelinek-Raviv) based map decoding algorithm is used in the proposed scheme. The Map algorithm is explained using three steps. i.e.

Forward Pass – Calculation (α):

These calculations made at each time interval k, for the simple four state RSC code [6] with trellis connectivity defined by the generator polynomial G = {7, 5}. First the trellis traversed in the forward direction at each node current state probability, α is calculated by multiplying the state probability at the previous node α_{k-1} (m') by the branch transition probability, Υ_{k-1}(m',m) given the received code pair R_k= {Y_{ks}, Y_{kp}}. This is expressed as follows:

$$\alpha_k(\mathrm{m}) = \frac{\sum m'\sum_{i=0}^{1} \gamma_i((y_k^s, y_k^p), m', m)\alpha_{k-1}(m')}{\sum m\sum m'\sum_{i=0}^{1} \gamma_i((y_k^s, y_k^p), m', m)\alpha_{k-1}(m')}$$

Where m is the current state, m' is the previous state and i is the data bit (‘0’ or ‘1’) corresponding to each branch exciting a node.

Backward Pass-Calculation (β):

Then the trellis is traversed in the reverse direction again the probability of each branch being taken is calculated. The current state probability β_k(m) is found by multiplying the probability of arriving in the previous state β_{k+1} (m) by the probability of taking the current state transitionΥ_{k+1}(m',m), [8] given the current received values R_{k+1} = {$Y_{K+1}{}^s$, $Y_{K+1}{}^p$} this is expressed as follows:

$$\beta_{k.}(\mathrm{m}) = \frac{\sum m'\sum_{i=0}^{1} \gamma_i((y_{k+1}^s, y_{k+1}^p), m', m)\beta_{k+1}(m')}{\sum m\sum m'\sum_{i=0}^{1} \gamma_i((y_{k+1}^s, y_{k+1}^p), m', m)\alpha_k\,(m')}$$

Where the symbols have the same meaning as before, but β is the backward state probability the transition probability for each branch between nodes is given by the equation:

$$\Upsilon_i\,((Y_k{}^s, Y_k{}^p), m', m) = P\,((Y_k{}^s, Y_k{}^p)\,|d_k = i, m', m)$$
$$.q\,(d_k = i|m', m)$$
$$.\pi\,(m|m')$$

P (…) is the transition probability of the channel; that is, the probability that a given received symbol $\{y_k^s, y_k^p\}$, will result when symbol $\{x_k^s, y_k^p\}$ is transmitted. This function is defined by the pdf of the channel; for example, a Gaussian pdf in the case of the AWGN channel. q (…) is the probability that any branch from a node can be taken, given the previous state m’, current state m and the data bit i is associated with the branch. q (…) is either ‘0’ or ‘1’, depending on the generator polynomial of the RSC encoder. π (m|m') represents the a priori information which forms the input to each component MAP decoder from the other MAP decoder, within the iterative decoding process

Calculation of Log Likelihood Probabilities, Λ (d_k):

Finally, the forward and backward probabilities at each time interval k of the trellis are used to provide a soft estimate of whether the transmitted data bit d_k was a ‘1’ or a ‘0’,for the RSC code with generator polynomial G = {7, 5}. The soft estimate is represented as a log likelihood ratio (LLR), [9] Λ (d_k) as this is a convenient form for representing a probability which can have a wide dynamic range it is calculated as follows:

$$\Lambda(d_k) = 1\mathrm{n}\,\frac{\sum m\ \sum m'\Upsilon 1\left((y_k^s, y_k^p), m', m\right)\alpha_{k-1}(m')\beta_k(m)}{\sum m\ \sum m'\Upsilon 0\left((y_k^s, y_k^p), m', m\right)\alpha_{k-1}(m')\beta_k(m)}$$

Λ (d_k) represents the probability that the current data bit is a ‘0’ (if Λ (d_k) is negative) or a ‘1’ (if Λ (d_k) is positive). After a number of iterations, typically 8...18, the de-interleaved value of Λ (d_k) from DEC_2 is converted to a hard decision estimate, d_k, of the transmitted data bit. This forms the output of the final turbo decoder stage.

The MAP algorithm, as described in part so far, is optimal for estimating the maximum likelihood data sequence on a bit-by-bit basis. However, the MAP algorithm is usually implemented in our paper even it is computationally complex.

III. RESULTS

Simulation results for four state, rate ½ turbo encoder using MAP decoding algorithm is shown below for different iterations, at different E_b/N_0 values. Filters like wiener and median filters are also used

Table 1: Comparison of PSNR Values for different Images at E_b/N_0= 1dB & 3dB for 4 iterations

Type of image	**PSNR values at Eb/No=1dB**	**PSNR values at Eb/No=3dB**
Noisy bit plane image	54.789891	55.805822
Turbo processed image	61.221192	69.348005
Turbo processed + Wiener filtered image	64.612813	69.825911
Turbo processed + Median filtered image	67.844165	70.795228

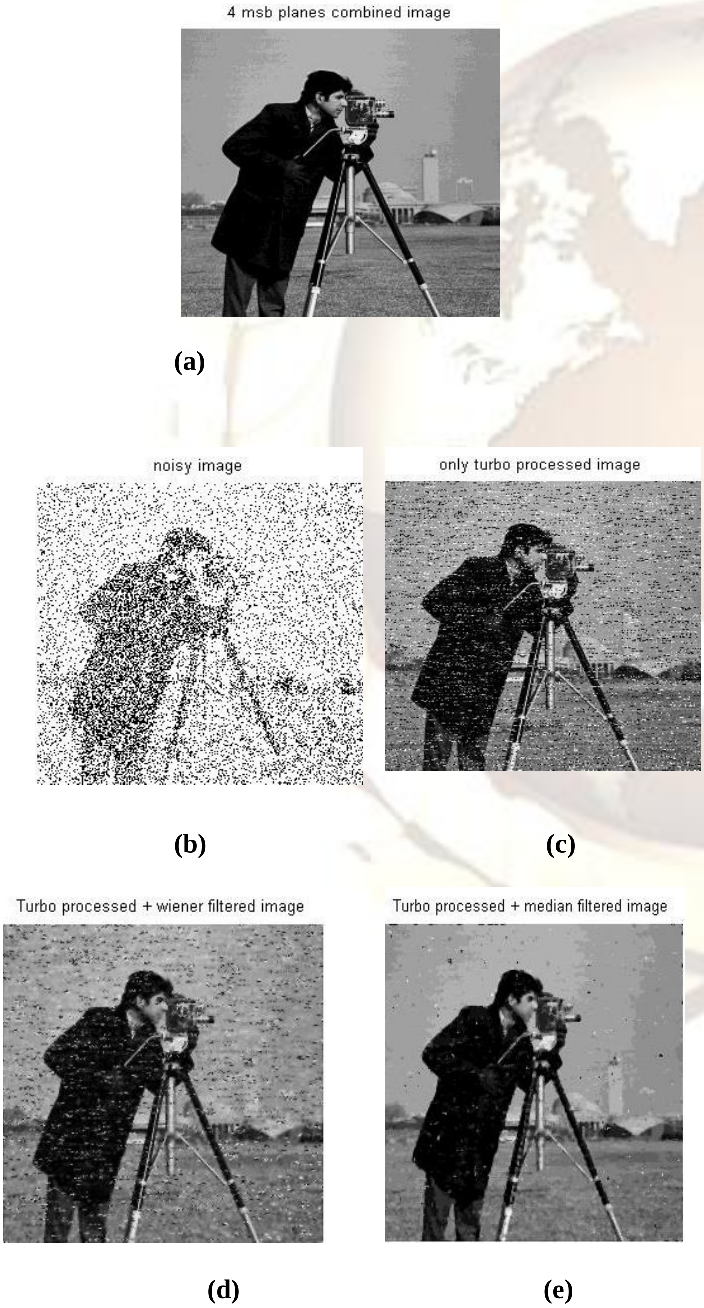


Fig. 4: Simulation result for G = (7, 5) rate ½ turbo Code for 4 iterations at $E_b/N_0 = 3$dB

(a) 4 MSB Planes combined image
(b) Noisy image
(c) Only Turbo Processed Image
(d) Turbo Processed with wiener filtered image
(e) Turbo processed with median filtered image

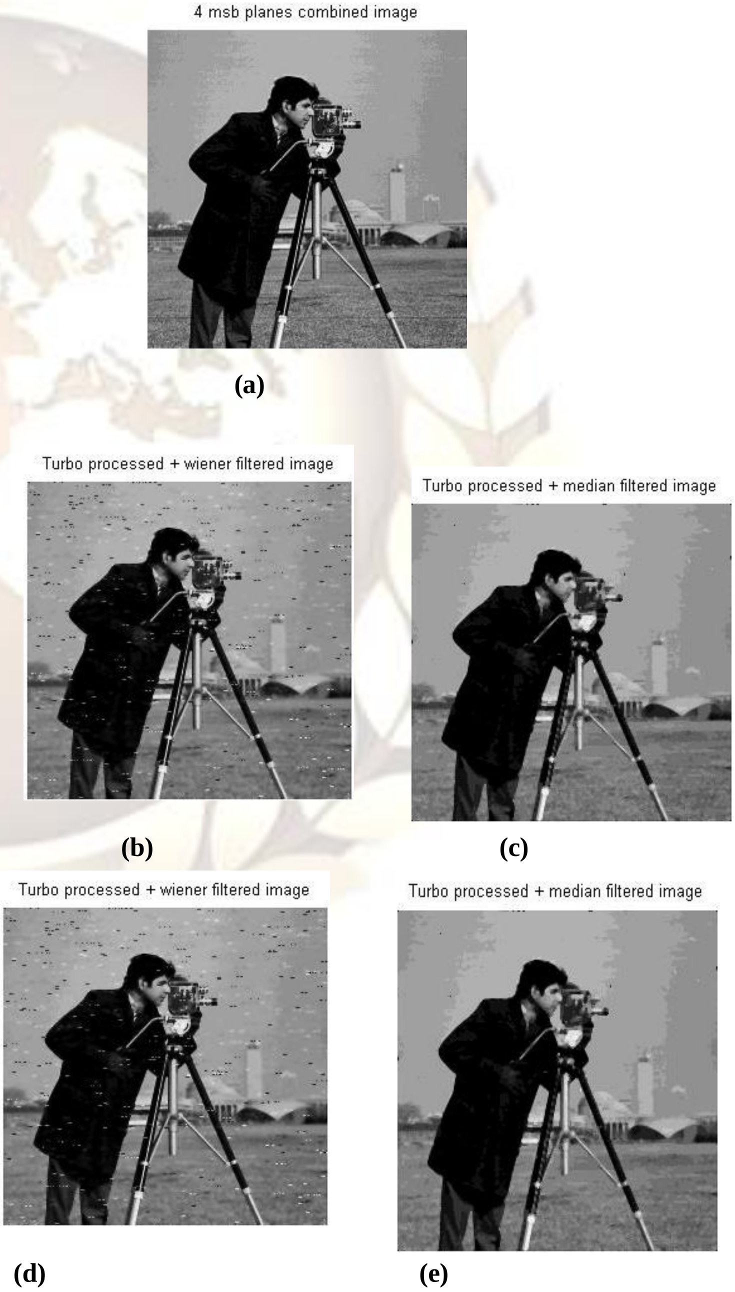
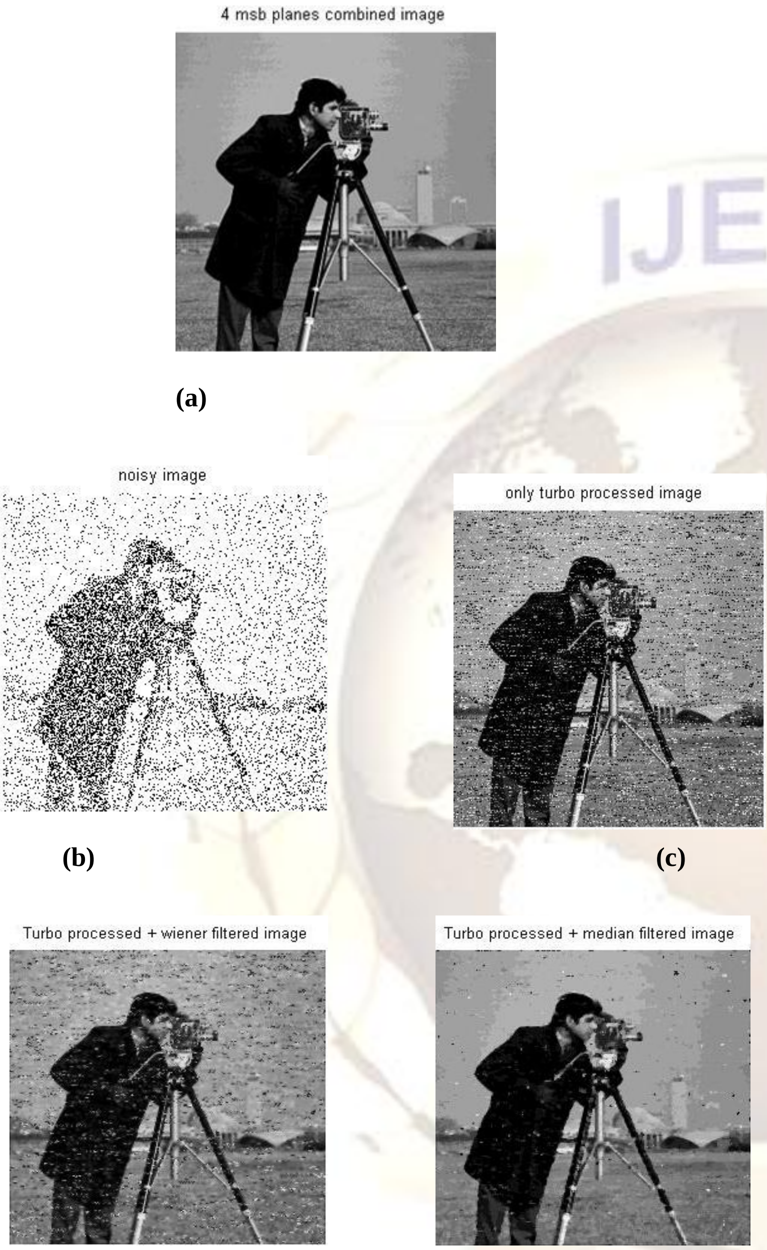


Fig. 5: Simulation result for G = (7, 5) rate ½ turbo Code for 4 iterations at $E_b/N_0 = 3$dB

(a) 4 MSB Planes combined image
(b) Noisy image
(c) Only Turbo Processed Image
(d) Turbo Processed with wiener filtered image
(e) Turbo processed with median filtered image


4 msb planes combined image

(a)

noisy image

(b)

only turbo processed image

(c)

Turbo processed + wiener filtered image

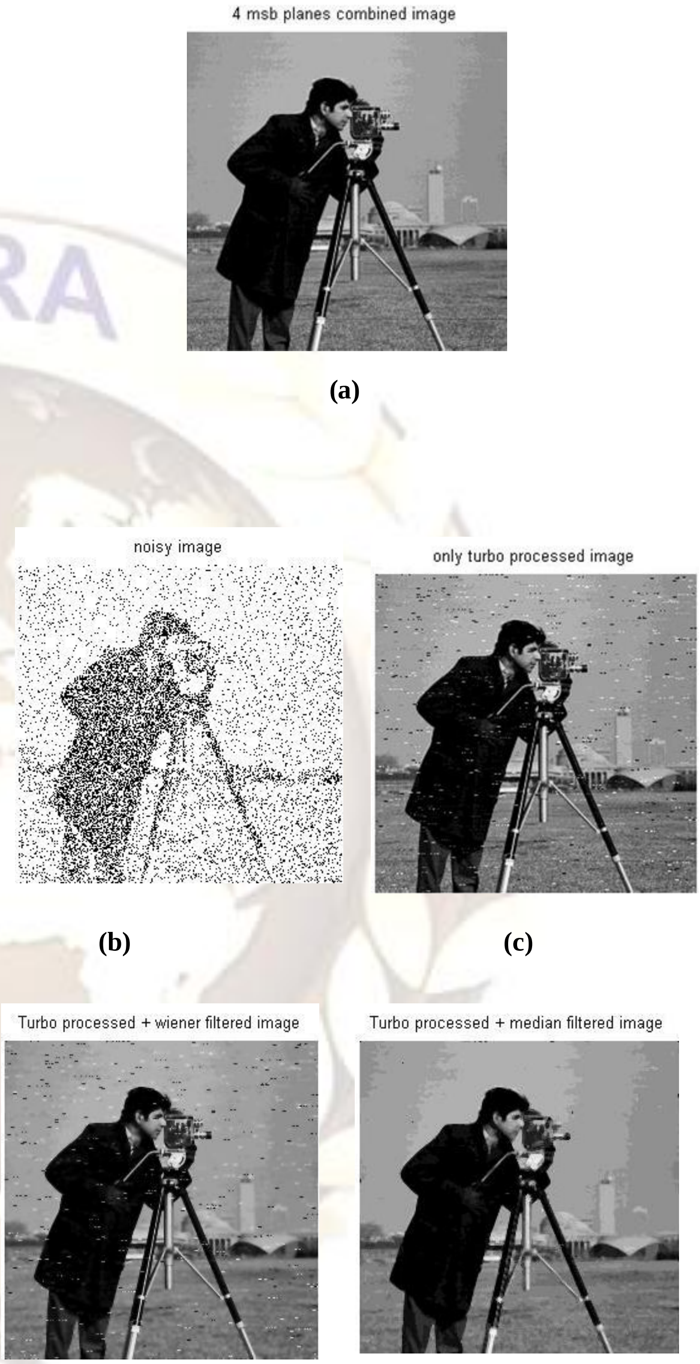
(d)

Turbo processed + median filtered image

(e)

Fig.6: Simulation result for G= (7, 5) rate ½ turbo code for 8 iterations at E_b/N_0=1dB

(a) 4 MSB Planes combined image
(b) Noisy image
(c) Only Turbo Processed Image
(d) Turbo Processed with wiener filtered imageTurbo processed with median filtered image


4 msb planes combined image

(a)

noisy image

(b)

only turbo processed image

(c)

Turbo processed + wiener filtered image

(d)

Turbo processed + median filtered image

(e)

Fig.7: Simulation result for G= (7, 5) rate ½ turbo code for 8 iterations at E_b/N_0=3dB

(a) 4 MSB Planes combined image
(b) Noisy image
(c) Only Turbo Processed Image
(d) Turbo Processed with wiener filtered image
(e) Turbo processed with median filtered mage

Table 2: Comparison of PSNR Values for different Images at E_b/N_0= 1dB & 3dB for 8 iterations

Type of image	**PSNR values at Eb/No=1dB**	**PSNR values at Eb/No=3dB**
Noisy bit plane image	54.789891	55.805822
Turbo processed image	61.287012	69.573931
Turbo processed + Wiener filtered image	64.617364	69.971618
Turbo processed + Median filtered image	67.981805	70.853381

CONCLUSION

The images being transmitted over noisy channels are extremely sensitive to the bit errors, which can severely degrade the quality of the image at the receiver. This necessitates the application of error control codes in the image transmission. This study presents an efficient image transmission by means of a new proposed method which takes the advantage of the superior performance of error control codes, i.e. turbo codes and de-noising filters. For comparison PSNR values of noisy, turbo, turbo-processed with wiener filtered and turbo processed with median filtered images are evaluated and results are compared. Using turbo coding it reconstructs the image properties in an effective manner, by considering the pixel neighborhood relations but it does not remove the noises introduced during the transmission. For removal of these noises, wiener and median filters are used here. Hence it is concluded that the turbo processed with median filtered image gives better performance than the other images.

REFERENCES

1. C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding: Turbo codes,” IEEE International Conference on Communications, Geneva, Switzerland, May 1993, pp. 1064-1070.
2. Erison Gose, Kenan Buyukatak, Onur Osman, and Osman N. Ucan “Image Transmission via Iterative Cellular- Turbo System”, World Academy of Science, Engineering and Technology, pp. 821-828, August 2010.
3. N. V. Boulgouris, N. Thomos, and M. G. Strintzis, “Image transmission using error-resilient wavelet coding and forward error correction,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Rochester, NY, Sept. 2002, pp. 549–552.
4. N. V. Boulgouris, D. Tzovaras, and M. G. Strintzis, “Lossless image compression based on optimal prediction, adaptive lifting and conditional arithmetic coding,” IEEE Trans. Image Processing, vol. 10, pp. 1–14, Jan. 2001.
5. G. Davis and J. Danskin, “Joint source and channel coding for image transmission over lossy packet networks,” in Proc. SPIE, vol. 2847, Apr.1996, pp. 376–387.
6. B. A. Banister, B. Belzer, and T. R. Fisher, “Robust image transmission using JPEG2000 and turbo codes,” IEEE Signal Processing Lett., vol. 9, pp. 117–119, Apr. 2002.
7. Gurmeet Kaur and Rupinder Kaur,“Image De-noising using Wavelet transform and Various Filters”, International Journal of research in Computer Science, Vol.2, Issue 2, pp15-21, February 2012.
8. Berrou, C. and Glavieux, A., “Near Optimum Error Correcting Coding and Decoding: Turbo-Codes,” IEEE Transactions on Communications, vol. 44, no. 10, pp. 1261-1271, October 1996.
9. Lin-Nan Lee, Fellow, IEEE, A. Roger Hammons, Jr. Feng-Wen Sun, and Mustafa Eroz, “Application and Standardization of Turbo codes in Third-Generation High-Speed Wireless Data Services,” IEEE Transactions on Vehicular Technology, Vol. 49, no. 6, November 2000.
10. Hamid R. Sadjapour, AT&T Research-Shannon Labs, Florham Park, NJ, “Maximum A Posteriori Decoding Algorithms For Turbo Codes”