IJCTA, 9(9), 2016, pp. 3873-3881

Gold Washing Adaptive Vector Quantization with Wavelet Transform for Colour Image Coding

R. Punidha[1], W. Nancy[2], C. Gunasundari[3] and D. Ruban Thomas[4]

ABSTRACT

In this paper an efficient color image compression technique using Adaptive Vector quantization with Discrete Wavelet Transform is proposed. DWT is used for depicting a image into various degree of resolution most commonly image is divided into various subbands. In general Vector quantization techniques that employ static code books must be large to maintain coding fidelity. This in turn, entails more bits for coding the labels and increases the coder/decoder complexity. To overcome this problem, in the proposed work an adaptive vector quantization technique with DWT is proposed in this work. In our proposed work, dynamic code book is generated where images outside database can also be compressed effectively. In the decoding phase, from the stored/transmitted codebook and index map an image is reconstructed. A new adaptive vector quantization based on DWT has been proposed for color images. This technique combines the features of both DWT coding and adaptive vector quantization. The universal codebook and proposed coding are compared.

Keywords: Colour Image compression, Wavelet Transform, Adaptive Vector Quantization, Gold Washing Method.

1. INTRODUCTION

Digital images while transmitting through internet consumes large amount of storage space and bandwidth. Storage and bandwidth can be effectively reduced by removing redundant pixels with the help of data compression[1]. Data compression is the process of converting original image into minute bits for efficient storage and transmission. there are two types of digital image compression techniques: 1.lossy compression 2.lossless compression[8]. In the case of lossy compression without affecting the the visual effect, the compressed images are recovered. In the case of lossless compression the original images can be recovered without the loss of information. The wavelet signal compression is a technique in which the input signals are expressed with a sum of power terms for wavelet function [7]. For the purpose of reducing the bit rate effectively, adaptive vector quantization is used which reduces bit rate less than one bit per pixel[2]. In this adaptive Vector Quantization, the image to be compressed are divided into blocks of size 4×4 pixels[5]. There are three levels in Adaptive Vector Quantization of images: 1. Dynamic Codebook Generation, 2. Image Encoding 3. Image Decoding. This paper presents a hybrid technique of Combining Discrete Wavelet transformation and adaptive Vector Quantization to compress color images[9].

The Adaptive Vector Quantization system dynamically updates the codebook according to the change in the input source. Depending on the local statistics of the input image a new codebook is generated[2]. Here the set of training vectors and the resultant codebook are adapted to the any input image outside the database. For each input image, a subset of vectors formed from the image is chosen as the training set for

[1] Professor/ IT, Vel Tech Multi Tech, *Email: punidhar@gmail.com*

[2] Assistant Professor/ECE, Jeppiaar Institute of Technology,

[3] Assistant Professor/CSE, Roever College of Engineering & Technology,

[4] Assistant Professor/ECE, Vel Tech Multi Tech,

codebook generation. Thus, the input image is better represented and for a given distortion, a smaller codebook may be needed.

Wavelets are set of basic functional unit derived by applying Wavelet transform. By the process of shifting and dilating wavelets are obtained from single prototype wavelet called mother wavelet. The flexible and highly efficient method of subband decomposition of images is developed using DWT[4]. In DWT, specific wavelet signals are the core of concentration of signal energy. This characteristic is useful for compressing images. Pixel block type of storage is ineffective when compared to storing images as series of wavelet. Wavelets have rough edges, by eliminating the blocking artifacts they are able to render better pictures. In DWT, using digital filtering techniques time-scale representation of the digital signal is obtained.

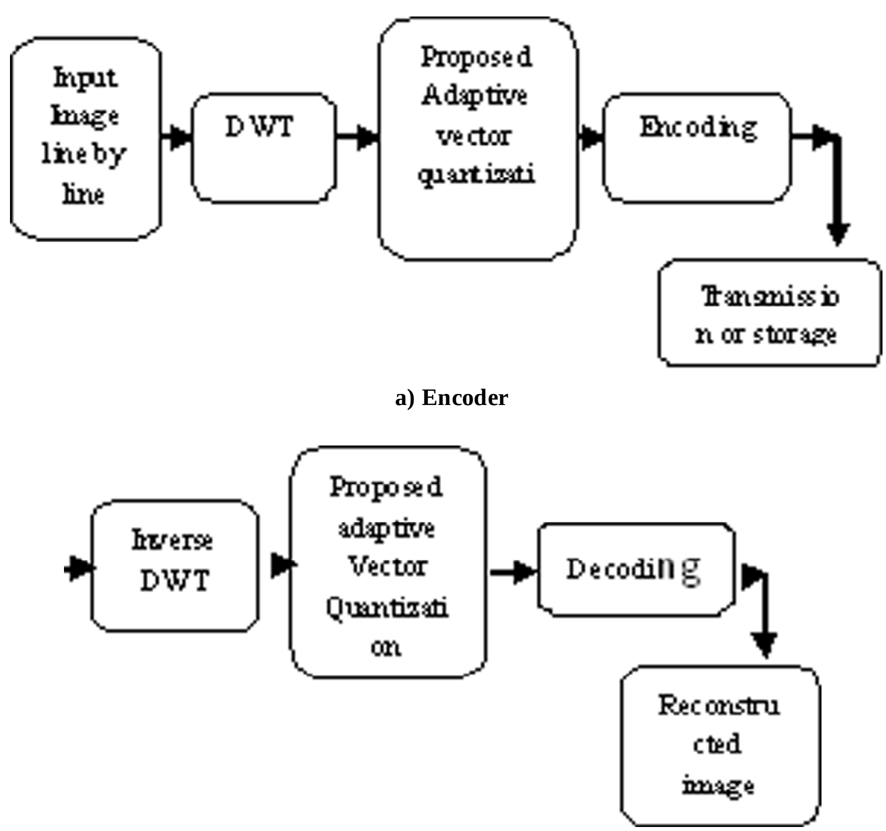
2. PROPOSED ADAPTIVE VECTOR QUANTIZATION WITH WAVELET TRANSFORM

In the proposed adaptive vector quantization encoding process, the image training set and the codebook should have the ability of adapting to each input image. Depending on the local statistics of the input image a new codebook is generated. Here the set of training vectors and the resultant codebook are adapted to the input image. For each input image, a subset of vectors formed from the image is chosen as the training set for codebook generation. Thus, the input image is better represented and for a given distortion, a smaller codebook may be needed. The proposed encoding and decoding process is shown in figure 1.

An adaptive vector quantization be expressed as follows:

$$Q_t : R^k \rightarrow C_t$$

where k is the dimension of code vector and C_t denotes a codebook that varies with time. The proposed wavelet transform coefficients based adaptive vector quantization uses the Gold Washing technique for



a) Encoder

b) Decoder

Figure 1: Proposed Adaptive Encoder and Decoder

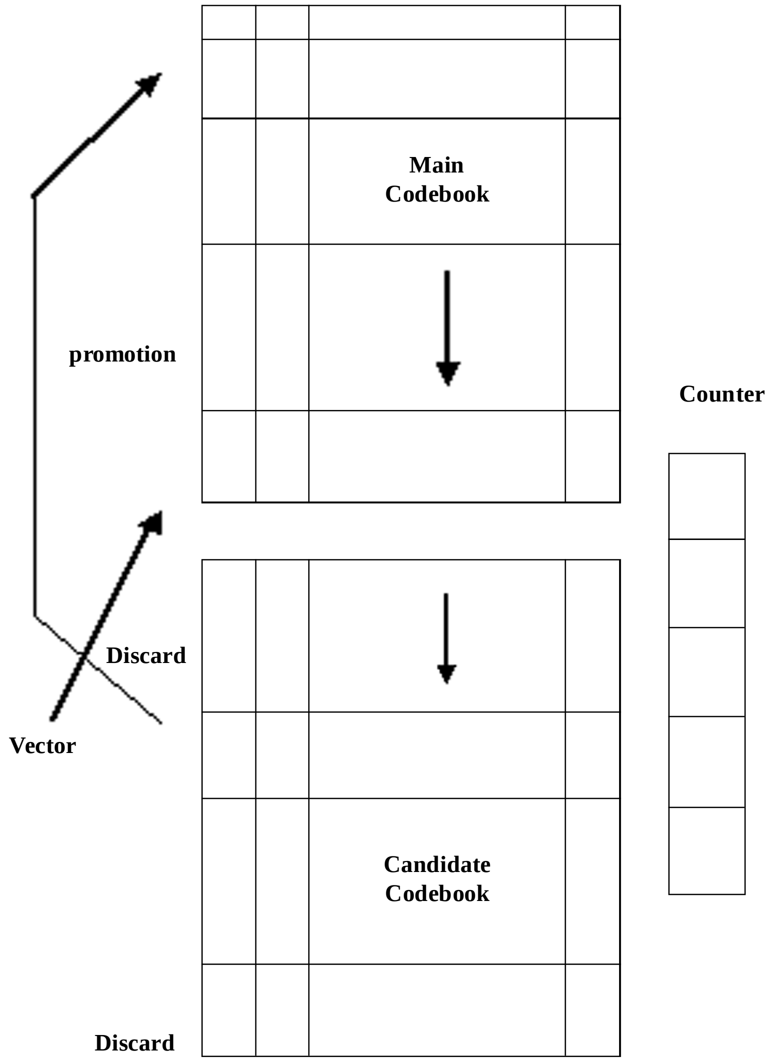


Figure 2: Proposed Adaptive Vector Quantization with adaptive wavelet transform coefficients

generating dynamic code book and the block interpolation method for creating the new code vector. The basic idea behind the proposed technique is to use a codebook based on wavelet transform coefficients and to keep on replacing unused or infrequently used code vectors in the encoding process as shown in figure 2. New code vectors are created when a close match is not found in the encoding process using block interpolation technique. This technique requires a series of table look-up operations similar to universal vector quantization. However, when none of the code vectors in the codebook is close enough to the input vector, i.e. when the minimum distortion is larger than the threshold, *TH*, the newly generated input vector is inserted into codebook. In the proposed adaptive vector quantization, when a new code vector is inserted, an old one has to be discarded simultaneously to maintain the original size of the codebook. This codebook updating process will increase the similarity between the code vectors in the code book and the vectors of the input image. Thus, the quality of reconstructed signals can be maintained

The proposed adaptive vector quantization starts by partitioning the original image of size $(R \times C)$ into non-overlapping sub-blocks of size $(n \times n)$ and mapping them to the frequency domain by applying the wavelet based transformation and obtain the transform coefficients β'_{ij} $\{i, j = 0, 1, 2, \ldots, n-1\}$. The process that is mentioned formerly is repeated to form the training vectors, $T_i (i = 0, 1, 2 \ldots)$ of all the partitioned sub-blocks. For a fixed rate $R > 0$, we employ a mechanism for adaptively changing a codebook C_n, which consists of two sub-codebooks namely codebook-1 C_n^1 and codebook-2 C_n^2, each of which contains $\lfloor 2^{nR} \rfloor$

discrete wavelet transform coefficients β'_{ij}. The operation of good performance the proposed codebook construction is shown in figure 1 C_n^1 aims at gathering codewords which have in the past test period and C_n^2 provides candidates for such good codewords. Each candidate in the C_n^2 will be tested for a period of length of $\lfloor 2^{nR} \rfloor$. Each part of the codebook contains the code vectors and the number of times they have been used. Initially, both parts could be empty or C_n^1 may be initialized to an existing codebook as in static quantization. However, each part of the codebook may be changed as input is quantized.

The distortion is calculated for each incoming training vector T_i between the input and each of the code vectors using $D = \sum_i d\ (x_{i\,,}\, Q\ \ (x_i))$, where $Q(x_i)$ are the code vectors, d is the Euclidean distance and x_i is the input vector. If the distortion D is within a pre-set threshold, *TH*, then the closest code vector is used to quantize the input and its frequency counter is incremented by 1. If the closest code vector comes from C_n^1, it is moved to the top of C_n^1 and other code vectors are pushed down by one notch. If the smallest distortion for an input training vector exceeds the allowable threshold *TH*, a new code vector is created by the block interpolation method, to match the input and is sent to the receiver. In block interpolation technique, data are selected from the blocks of source information and are quantized. The quantized data are then used to find the neighbouring data values by the interpolation method. The coefficient codes for the neighbouring data are also constructed. The quantized level and jumping step for the coefficient table are selected according to performance requirements. If a (4 × 4) sub-image block is used, the four corner pixels are quantized and the others are estimated by interpolation using these 4 quantized values. After quantization and interpolation, let the estimated sub-image block be

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Here, a_{00}, a_{03}, a_{30} and a_{33} are idividually quantized from the original color values of the four corner pixels. The linear interpolation formula for the other pixels is described as follows,

$$a_{ij} = b_j\, a_{i0} + b_{3-j}\, a_{i3} + b_i\, a_{0j} + b_{3-i}\, a_{3j}, \tag{1}$$

where $(i, j) \in \{(0, 1), (0, 2), (1, 0), (2, 0), (1, 3), (2, 3), (3, 1), (3, 2)\}$ and $b_0 = 0$, $b_1 = \frac{2}{3}$, $b_2 = \frac{1}{3}$ and $b_3 = 0$.

According to the values of $a_{01}, a_{02}, a_{10}, a_{20}, a_{13}, a_{23}, a_{31}$ and a_{32}, the values of a_{11}, a_{21}, a_{12} and a_{22} are estimated by the following formula,

$$a_{i,j} = \bar{b}_i a_{0j} \div \bar{b}_{3-i} a_{3j} + \bar{b}_j a_{i0} + \bar{b}_{3-j} a_{i3},$$

for $$1 \le i, j \le 2 \tag{2}$$

where $\bar{b}_1 = \frac{1}{3}$, $\bar{b}_2 = \frac{1}{6}$.

This new code vector is then placed on the top of C_n^2, and its initial frequency count is set to 0. All the previous entries of C_n^2 are pushed down by one notch. As a result, there could be an entry that is pushed out of the bottom of C_n^2. The frequency count of the code vector thus pushed is compared with a preset value. If the frequency count of this entry is above the preset value, the entry is placed in the top of C_n^1, else it is discarded; otherwise, and all the entries in C_n^1 are pushed down. In such a case, if there is an entry at the bottom of C_n^1, it would be discarded.

In the above process, C_n^2 serves as a testing area for any new code vector that is created. Entries in C_n^2 are stored in the order in which they arrive. If a new entry travels through the entire area and has not been used often enough as determined by the frequency threshold, then it is removed. Otherwise, the entry is considered useful and is moved to C_n^1 where the code vectors are arranged according to how recently they have been used. The entry at the bottom to be replaced is the least recently used and is considered the least useful. Thus, the system works by retaining the potentially most useful code vectors and by removing the least useful ones to maintain the constant code book size. In general, C_n^1, functions as a frame adaptive buffer and C_n^2 functions as a block adaptive buffer. The sizes of, C_n^1, and C_n^2 can be adjusted according to data statistics.

Upon receiving data, the decoder checks the identification bit first. If it is type1, the code vector index is reconstructed with the received code by using the entropy decoder. The approximated transform coefficients β'_{ij} are then retrieved by performing a table lookup with the aid of the code vector index and the codebook. Otherwise, the transform coefficients β'_{ij} are produced from quantized sample data and a coefficient table and the codebook is updated. Finally these transform coefficients are subjected to inverse transform with the help of basis functions of the proposed vector quantization to get back the decompressed image. The steps involved in the proposed encoding algorithm, is presented as an algorithm

PROPOSED ALGORITHM

INPUT : Color Image

OUTPUT : Compressed Image

Encoding Steps

1. Partition the input image [I] into non-overlapping image regions of size ($n \times n$).
2. Repeat the steps 3 to 4 for all the image regions.
3. Compute the Discrete wavelet transform coefficients [β']
4. Form a training vector T_i using the wavelet transformed coefficients.
5. Create the initial codebook from a random generator.
6. Encode the incoming vector x_i by code vector matching. If a best matched code vector y_j exists such that distortion D is smaller than TH, then an identification bit of type 1 and corresponding index value j are identified. The codebook is updated by the Gold-Washing method.

7. If no code vectors meet minimum distortion requirement, the block-data interpolation is applied to generate a new code vector. This new code vector is added to codebook according to the Gold-Washing method. An identification bit of type 2, x_i is encoded using coefficient table and quantized sample data and index values are identified.
8. The index values along with identification type are subjected to entropy coding and the coded value is transmitted to the receiver through channel

DECODING STEPS

1. At receiving end, decode the index values and identification type. Then form an approximation to original wavelet transform coefficients [β′] using index values and identification type
2. Reconstruct the input image region [I] using the Wavelet transform functions
3. Repeat the steps 1 to 2 until all the image regions [I] are reconstructed

3. PERFORMANCE ANALYSIS

The performance of the proposed vector design on vector transform coefficients based transform coding is reported by calculating the value of peak signal-to-noise ratio (PSNR), which is defined as

$$PSNR = 10\log_{10}\left[\frac{255}{e_{ms}^{\;2}}\right]^2$$

where the average mean-square error e_{ms}, is

$$e_{rms}^2 = \frac{1}{NM}\sum_{i=1}^{N}\sum_{i=1}^{M} E\left(u_{i,j} - u'_{i,j}\right)^2$$

where $\{u_{i,j}\}$ and $\{u'_{i,j}\}$ represent the $N \times M$ inputl and reconstructed images respectively.

4. EXPERIMENTS AND RESULTS

The discrete wavelet transform based adaptive vector quantization has been experimented with more than 1000 test images, having different low level primitives. For illustration two test images Lena and Pepper, both of size (128 × 128) with gray scale values in the range (0–255) are shown in figure 3(a) and 3(b) respectively. The input images are partitioned into various non-overlapping sub-blocks of size (4 × 4). We then apply the proposed discrete wavelet based transformation as described in section 2.0 and obtain the transform coefficients β'_{ij}. The sum of the codebook-1, C_n^1 size and the codebook-2, C_n^2 size is set to 512. The individual sizes of C_n^1 and C_n^2 are selected according to image statistics. The frequency threshold is set to 2 for a moderate C_n^2 size. It makes the code vectors in C_n^2 easy to survive and migrate to C_n^1. The initial codebook is created from a random number generator. The distortion threshold is chosen such that the performance of reconstruction can be similar to the results from the Simulated Annealing method. Here, the distortion threshold of C_n^1 is the same as that of C_n^2.

The transformed block [β′] is then encoded. The resultant index values are entropy coded and transmitted to the receiver. At the decoder, the index values are decoded and using Gold Washing technique the identification type the approximated transformed block [β′] is generated. Then the decompressed original image is reconstructed with the wavelet transform coefficients. The bit per pixel (bpp) scheme is used to

(a) lena **(b) pepper**

Figure 3: Original test images considered for the proposed AVQ

(a) lena **(b) pepper**

Figure 4: Results of proposed AVQ when bpp = 0.28

(a) lena **(b) pepper**

Figure 5: Results of vector quantization with discrete wavelet transform using algorithm when bpp = 0.25

Table 1
PSNR values obtained with the proposed adaptive vector quantization, vector quantization using Simulated Annealing algorithm for different bpps.

Bit Rate	*Proposed AVQ*		*Simulated Annealing Algorithm*	
	Lena	*Pepper*	*Lena*	*Pepper*
0.28	33.17	33.48	33.20	33.21
0.21	30.99	31.23	30.17	30.93
0.19	29.95	30.16	29.55	29.89
0.18	29.64	29.83	29.14	29.49
0.16	29.12	29.26	28.69	28.99
0.14	28.66	29.18	27.6	28.17

estimate the transmission bit rate. The performance of the proposed scheme is measured with the standard measure Peak-Signal-to-Noise-Ratio (PSNR). We obtain PSNR values of 33.51dB and 33.56dB for a bit rate of 0.28 for the input images 1(a) and 1(b) respectively and the corresponding resulting images are shown in figures 4(a) and 4(b) respectively. Similarly, we obtain PSNR values of 33.51dB and 33.56dB for a bit rate of 0.25 and the corresponding resulting images are shown in figures 5(a) and 5(b) respectively. The experiment is repeated by varying the bpp for all the 1000 images and the results for the Lena and Pepper images are presented in table 1.

In order to measure the efficiency of the proposed discrete wavelet transform based adaptive vector quantization, we conduct experiments with proposed algorithm on discrete wavelet transform coefficients. In the Simulated Annealing algorithm, the transform coefficients of the original image are used to generate the codebook, and then this codebook is utilized to encode and decode the original image. The experiments are conducted for different bpp and the corresponding PSNR values obtained are incorporated in the same table 1 for both the input images and the corresponding results for 0.28 bpp are shown in figure 3(a) and 3(b) respectively. It is evident that the proposed adaptive vector quantization outperforms vector quantization based on Simulated Annealing algorithms. The error contour of the difference image using the Simulated Annealing algorithm is clearer than that using the adaptive vector quantization method. In other words, edge information is well preserved with the proposed adaptive vector quantization method since most edges are estimated with the block-data interpolation method.

5. CONCLUSION

This paper proposes a new adaptive vector quantization based on Discrete Wavelet Transform for color image coding. This technique combines the features of both discrete wavelet transform and adaptive vector quantization. In the proposed work wavelets are used to extract fine details of image. Bit rate is further reduced by applying VQ. Compared to JPEG scheme, the only complex operation in the algorithm is the compression of images outside the codebook, which is improved by designing a adaptive codebook. The code book is designed with Gold-Washing technique on discrete wavelet transform coefficients. This combination has the features of good local adaptivity, low complexity and significant compression ratio. The Gold-Washing method can reach rate distortion function for memory less sources. The performance of the proposed work is measured with standard PSNR value and is compared with that of the universal codebook. The performance is encouraging and promising to suit the current requirements.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the contribution of Govt. of India for Financial Assistance, DST-FIST F.NO:SR/FST/College-189/2013.

REFERENCES

[1] E. A. B. Da Silva; D. G. Sampson; M. Ghanbari, "A successive approximation vector quantizer for wavelet transform image coding", IEEE Transactions on Image Processing Vol. 5, Issue: 2, Feb. 1996.

[2] Srilakshmi Inuganti, Veerraju Gampala, "Image Compression using Efficient Codebook Generation" ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering Vol. 5, No. 2, February 2015.

[3] Yunhao Bai, "Image Compression Based on Discrete Cosine Transform and Multistage Vector Quantization" International Journal of Multimedia and Ubiquitous Engineering Vol. 10, No. 6, p. 347-356, 2015.

[4] M. Mozammel Hoque Chowdhury and Amina Khatun, "Image Compression Using Discrete Wavelet Transform" IJCSI International Journal of Computer Science Issues, Vol. 9, No. 1, July 2012.

[5] Pamela C. Cosman, "Vector Quantization of Image Subbands", IEEE Transactions on Image Processing, Vol. 5, No. 2, February 1996.

[6] Priya Pareek, "An Image Compression Using Multilayer Wavelet Transform with 2DTCWT", International Journal of Computer Applications (0975–8887) Vol. 102, No. 1, September 2014.

[7] Hunny Sharma, Satnam Singh "Image Compression Using Wavelet Based Various Algorithms" International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 5, No. 5, May 2015.

[8] Azam Karami, "Lossy compression of hyperspectral images optimizing spectral unmixing", 2015.

[9] Zhang Ning and Zhu Jinfu, "Study On Image Compression And Fusion Based On The Wavelet Transform Technology", International Journal On Smart Sensing And Intelligent Systems vol. 8, No. 1, March 2015.

[10] R.A. Rutenbar, "Simulated annealing algorithms: an overview", IEEE circuits and devices magazine, Vol. 5, No. 1, Jan. 1989.

[11] P. Caenevali, "Image processing by simulates annealing", IBM journal of Research and Development, Vol. 29, No. 6, Nov. 1985

[12] Shaou-Gang Miaou, "A hardware-oriented gold-washing adaptive vector quantizer and its VLSI architectures for image data compression", IEEE Transactions on Circuits and Systems for Video Technology , Vol. 10, No. 8, Dec. 2000.

[13] S. Grgic, "Performance analysis of image compression using wavelets", IEEE Transactions on Industrial Electronics, Vol. 48, No. 3, Jun. 2001.

[14] B. S. Manjunath, "Texture features for browsing and retrieval of image data", IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol. 18, No. 8, Aug. 1996.

[15] Jong-Sen Lee; Naval Research Laboratory, Washington, DC 20375, "Digital Image Enhancement and Noise Filtering by Use of Local Statistics", IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. PAMI-2, No. 2, March 1980.

[16] J. Portilla, "Image denoising using scale mixtures of Gaussians in the wavelet domain" IEEE Transactions on Image Processing, Vol. 12, No. 11, Nov. 2003.