Image Denoising Based Using Hybrid Techniques Mixed between (Hard and Soft Threshold) With Multiwavelet Transform and Multi-Stage Vector Quantization

أزالة الضوضاء من الصورة بأستخدام تقنيات هجينة مدموجة بين المدخل المادي والبرمجي مع تحويل متعدد المويجة ومكمم المتجه متعدد المراحل

Adil Abdulwahhab Ghidan Al-Azzawi

عادل عبد الوهاب غيدان العزاوي

Diyala University / Collage of science / Computer science Department

جامعة ديالى / كلية العلوم / قسم علوم الحاسوب

Abstract

The Image denoising naturally corrupted by noise is a classical problem in the field of signal or image processing. Denoising of a natural images corrupted by Gaussian noise using new techniques of multi-wavelet techniques depended on losing of some error are occurs in reconstruct in pre-processing of multiwavelet then the new multiwavelet are very effective because of its ability to reduce losing of some data in reconstruct. Multi-wavelet can satisfy with symmetry and asymmetry which are very important characteristics in signal processing. The better denoising result depends on the degree of the noise. Generally, its energy is distributed over low frequency band while both its noise and details are distributed over high frequency band. Corresponding hard threshold used in different scale high frequency sub-bands.In this paper proposed to indicate the suitability of different multi-wavelet based on using the mixing between Hard and Soft threshold that named as Hybrid threshold technical depended on the parts of the multi-wavelet decomposition, according the value of noise in the decomposition parts used the threshold techniques for example using the soft threshold on the two first parts LL and LH decompositions because that the amount value of pixels in this part

is Low frequency and some of Hard and then using the Hard threshold of the remaining two parts HL and HH part because the amount value of pixels is High frequency, then the performance calculation of image denoising algorithm in terms of PSNR value. Finally it's compare between multiwavelet traditional techniques Hard, Soft threshold and produced best denoised image using (Hybrid threshold) image denoising algorithm in terms of PSNR Values based on mixed thresolding (hard and soft thresolding) by using the first (Hard threshold) in LL and LH part and the second (soft thresolding) in HL and HH part from multiwavelet decomposition.

المستخلص

أزالة ضوضاء من صورة مشوهة طبيعياً بالضوضاء تعتبر مشكلة تقليدية في مجال الأشارة أو معالجة الصور.
أزالة الضوضاء من الصور المشوهة بالضوضاء الكاوسي بأستخدام تقنيات جديدة لتحويل الموجة المتعدد يشد على خسارة بعض الأخطاء التي يحدث عند أعادة البناء لمتعدد الموجة ولذلك فأن متعدد الموجة سيكون مؤثر لقابليته على تقليص خسارة البيانات عند أعادة البناء. متعدد الموجة له قابلية التناظر وعدم التناظر وهي خفية مهمة في معالجة الأشارة. ونتيجة لأزالة أفضل ضوضاء والذي يعتمد على درجة الضوضاء. بصورة عامة فأن طاقة الصورة ستتوزع على حزمة الترددات الواطئة ينما تتوزع الضوضاء والتفاصيل على حزمة الترددات العالية. طبقا لهذا فأن المدخل الحاد تم أستخدامه مع مختلف الحزم الترددية العالية.
في هذا البحث المقترح تمت الأشارة متعدد الموجة المناسب بالأعتماد على الدمج بين المدخل المادي والبرمجي وكما مسمى المدخل المدمج الذي يعتمد على أجزاء من أعادة تكوين متعدد الموجة. طبقا لهذا فأن قيمة الضوضاء تم استخدامها قني تقنية المدخل فمثلا فأن المدخل البرمجي تم أستخدامه في الجزئين LL,LH والمدخل المادي تم أستخدامه مع الجزلين HH,HL ثم حساب الأداء لخوارزمية أزالة الضوضاء من الصورة دلالة PSNR . أخيرا تم مقارنة النتائج لمتعدد الموجة التقليدي مع متعدد الموجة المدمج المدخل بدلالة PSNR بأستخدام أولا LL,LH وثانيا HL,HH في أعادة تكوين متعدد الموجة.

Key Words: Discrete Multiwavelet Transform, hard threshold, soft threshold, PSNR peak signal to noise ratio, SNR signal to noise ratio, RMSE root mean square error.

Image Denoising Based Using Hybrid Techniques Mixed between (Hard and Soft Threshold) With Multiwavelet Transform and Multi-Stage Vector Quantization

Adil Abdulwahhab Ghidan Al-Azzawi

Introduction

Digital representation of image has created the need to transform the image from one domain to other that transformed used in digital image processing applications, observed image is modeled to be corrupted by different types of noise that results in a noisy version. Hence, image denoising is an important problem that aims to find an estimate version from the noisy image that is as close to the original image as possible, or to efficient compression algorithms that will reduce the storage space and the associated channel bandwidth for the transmission of images. Transform aims at changing the representation of a signal or a function by using of a mathematical operation. It is possible also to decompose a complex problem into simpler ones for obtaining simpler solution. Transforms play an important role in different signal processing applications like filtering, pattern recognition, restoration, spectrum estimation, signal enhancement, localization and compression [1]. The performance of each application depends on several factors, and hence, each application may need a different transform technique for a better solution [2].Multiwavelet is one type of the image transform that depended on the filter banks require a vector-value input signal. There are a number of ways to produce such a signal from 2-D signal image data. Perhaps the most obvious method is to use adjacent row and columns of the image data [3]. However, these approach doses not work well for general multiwavelets and leads to reconstruction artifacts in the lowpass data after coefficient quantization [4]. This problem can be avoided by constructing "constrained" multiwavelets, which possess certain key properties. Unfortunately, the extra constraints are somewhat restrictive; image compression test show that constrained multiwavelets do not perform as well as some other multifilters

[5]. Another approach is to first split each row or column into two half-length signal, and then use these two half signal as the channel inputs into the multifilter. A naïve approach, as Strela points out [6], is to simply take the odd samples for one signal and the even samples for the second signal.

This paper investigates the suitability a different number of different techniques have been for digital image denoising. The aim of these denoising techniques is to make as good an estimate as possible of the original image. There are different discrete transform methods to eliminate the amount of noise in images using discrete cosine transform, discrete wavelet transform, and discrete multiwavelet transform. These deals with the method of image denoising using multiwavelet transform. Discrete multiwavelet transform is a good tool for solving images processing problems such as image denoising. This research introduces how to generate a Gaussian noise in image depending on the signal-to-noise ratio (SNR), how to eliminate this type of noise from digital. The problem of Image de-noising can be summarized as follows. Let A(i,j) be the noise-free image and B(i,j)the image corrupted with independent Gaussian noise Z(i,j)

$$B(i,j)= A(i,j)+\sigma Z(i,j) \qquad ...(1)$$

Where Z(i,j) has normal distribution N(0,1) . The problem is to estimate the desired signal as accurately as possible according to some criteria. In the wavelet domain, if an orthogonal wavelet transform is used, the problem can be formulated as

$$Y(i,j)= W(i,j)+N(i,j) \qquad ...(2)$$

where Y(i,j) is noisy wavelet coefficient; W(i,j) is true coefficient and N(i,j) noise, which is independent Gaussian. In multi-wavelet aspects, the symmetry and dissymmetry of the wavelet is rather important in signal processing .But singlewavelets with orthogonal intersection and compact-supporting are not symmetric except Harr. Recently, research on multi-wavelet is an active orientation. As multiwavelet can satisfy both symmetry and asymmetry which are very important characters in signal processing. Multi-wavelet is commonly used in image compression, image de-noising, digital watermark and other signal processing field, so it is especially appropriate to processing complex images.

There are r compact-supporting scaling functions $\varphi=(\varphi_1, \varphi_2,\cdots. \varphi_r)^T$ and they are inter-orthogonal with the wavelet functions $\psi=(\psi_1, \psi_2,\cdots\cdots \psi_r)^T$, $\varphi r(t)(l=1,2,\cdots r)$. The orthogonal basis of $L_2(R)$ space is $2j/2\ \psi_r\ (2jt-k)(j,\ k\in Z,l=1,2,\cdots,r)$. Hk,, Gk are the N*N matrix finite response filters with orthogonal basis, then the following specific equations can be obtained:

$$\Phi(t)= \sqrt{2}\ \Sigma HK\ \Phi(2t-k) \qquad ...(3)$$

$$\psi(t)= \sqrt{2}\ \Sigma GK\ \psi(2t-k) \qquad ...(4)$$

Multiwavelet Theory

In multiwavelet transform, we use multiwavelet as transform basis. Multiwavelet functions are functions generated from one single function ψ by scaling and translation:

$$\psi_{a,b}(t)= \frac{1}{\sqrt{a}}\psi_a(\frac{t-b}{a}) \quad \ldots (5)$$

The mother wavelet $\psi(t)$ has to be zero integral, $\int \psi_{a,b}(t)dt = 0$. From (1) we see that high frequency multiwavelet correspond to $a > 1$ or narrow width, while low frequency multiwavelet corresponds to $a < 1$ or wider width. The basic idea of wavelet transform is to represent any function f as a linear superposition of wavelets. Any such superposition decomposes f to different scale levels, where each level can be then further decomposed with a resolution adapted to that level. One general way to do this is writing f as the sum of wavelets $\psi_{m,n}(t)$ over m and n . This leads to discrete wavelet transform [7]:

$$f(t) = \Sigma_{m,n}\ \psi_{m,n}(t) \quad \ldots (6)$$

By introducing the multi-resolution analysis (MRA) idea by Mallat [3], in discrete wavelet transform we really use two functions: wavelet function $\psi(t)$ and scaling function $\varphi(t)$. If we have a scaling function $\varphi(t) \in L^2(R)$, then the sequence of subspaces spanned by its scaling and translations $\psi_{j,k}(t)=2^{j/2}\ \varphi(2^j\ t - k)$, i.e

$$Vj = span\ \{\varphi j,k\ (t),j,k\ \in Z\} \quad \ldots (7)$$

Constitute a MRA for $L^2(R)$.

$\varphi(t)$ must satisfy the MRA condition [7]:

$$\varphi(t)=\sqrt{2}\sum h(n)\ \varphi(2t\text{-}2) \quad \ldots (8)$$

For $n \in Z$. In this manner, we can span the difference between spaces V_j by wavelet functions produced from mother wavelet: $\psi_{j,k}(t)=2^{j/2}\,\varphi(2^j t - k)$ Then we have [8]:

$$\psi_{j,k}(t) = \sqrt{2}\sum g(n)\ \varphi(2t\text{-}2) \qquad \dots (9)$$

For orthogonal basis we have [8]:

$$g(n) = (-1)^n\ h(-n+1) \qquad \dots (10)$$

If we want to find the projection of a function $f(t) \in L^2(R)$ on this set of subspaces, we must express it in e as a linear combination of expansion functions of that subspace [4]:

$$f(t) = \sum_{n=-\infty}^{\infty} c(t)\ \varphi(t) + \sum_{n=-\infty}^{\infty}\ \sum_{n=-\infty}^{\infty} d(t)\ \psi_{j,k} \qquad \dots (11)$$

Where $\varphi_k(t)$ corresponds to the space V_0 and $\psi_{j,k}(t)$ corresponds to wavelet spaces.By using the idea of MRA implementation of wavelet decomposition can be performed using filter bank constructed by a pyramidal structure of lowpass filters $h(n)$ and highpass filters $g(n)$ [9,10].

and wavelet functions satisfy the following two-scale dilation equations:

The Multiwavelet two-scale equations resemble those for scalar wavelets using Eqs.(3) and Eqs.(4). Note, however, that {H} and {G} is matrix filter, i.e., H and G are r × r matrices for each integer k, and the matrix elements in these filters provide more degrees of freedom than a traditional scalar wavelet. These extra degrees of freedom can be used to incorporate useful properties into the multileveled filters, such as orthogonally, symmetry, and high order of approximation. The key then is to figure out how to make the best use of these extra degrees of freedom. Multifilter construction methods are already being developed to exploit them. However, the multi-channel nature of Multiwavelet also means that the sub-band structure resulting from passing a signal through a multifilter bank is different. Sufficiently different, in fact, so that established quantization methods do not perform as well with Multiwavelet as they do with wavelets [2]. One famous Multiwavelet filter is the GHM filter proposed by Geronimo, Hardian, and Massopust [3]. The GHM bases offer a combination of orthogonality, symmetry, and compact support, which cannot be achieved by any scalar wavelet basis [8]. According to

Eq. (1) and (2) the GHM two scaling and wavelet functions satisfy the following two-scale dilation equations:

$$\begin{bmatrix} \phi 1(t) \\ \\ \phi 2(t) \end{bmatrix} = \sqrt{2}\ \Sigma_k H_K \begin{bmatrix} \phi 1(2t\text{-}k) \\ \\ \phi 2(2t\text{-}k) \end{bmatrix}$$

$$\begin{bmatrix} \psi 1(t) \\ \\ \psi 2(t) \end{bmatrix} = \sqrt{2}\ \Sigma_k G_K \begin{bmatrix} \psi 1(2t\text{-}k) \\ \\ \psi 2(2t\text{-}k) \end{bmatrix}$$

Where Hk for GHM system are four scaling matrices H0, H1, H2, and H3 [9],

$$H_0 = \begin{bmatrix} \frac{3}{5\sqrt{2}} & \frac{4}{5} \\ \frac{-1}{20} & \frac{-3}{10\sqrt{2}} \end{bmatrix},\ H_1 = \begin{bmatrix} \frac{3}{5\sqrt{2}} & 0 \\ \frac{9}{20} & \frac{1}{\sqrt{2}} \end{bmatrix},\ H_2 = \begin{bmatrix} 0 & 0 \\ \frac{9}{20} & \frac{-3}{10\sqrt{2}} \end{bmatrix},\ H_3 = \begin{bmatrix} 0 & 0 \\ \frac{-1}{20} & 0 \end{bmatrix} \quad \ldots (14)$$

Where Gk for GHM system are four wavelet matrices G_0, G_1, G_2, and G_3 [9],

$$G_0 = \begin{bmatrix} \frac{-1}{20} & \frac{3}{10\sqrt{2}} \\ \frac{1}{10\sqrt{2}} & \frac{3}{10} \end{bmatrix},\ G_1 = \begin{bmatrix} \frac{9}{20} & \frac{-1}{\sqrt{2}} \\ \frac{9}{10\sqrt{2}} & 0 \end{bmatrix},\ G_2 = \begin{bmatrix} \frac{9}{20} & 0 \\ \frac{9}{10\sqrt{2}} & \frac{3}{10} \end{bmatrix},\ G_3 = \begin{bmatrix} \frac{1}{20} & 0 \\ \frac{-1}{10\sqrt{2}} & 0 \end{bmatrix} \quad \ldots (15)$$

While earlier multiwavelet literature goes back further [7], some of the earliest developed multiresolution theory of multiwavelets can be found in a paper by Goodman. [10]. Strelas in his Ph.D. thesis [6] extends the theory of multiwavelets even further and presents it in terms of Pre-Processing multifilter banks in both time and frequency domains [1].

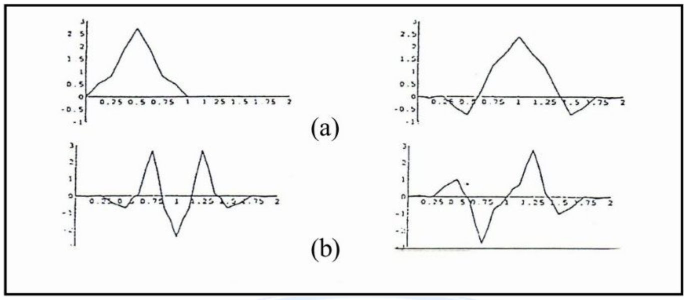

Figure (1): GHM pair of, (a) Scaling Functions, (b) Multiwavelets

Multi-Stage vector Quantization

Vector quantization is a powerful tool for data reduction. Vector quantization extends scalar quantization to higher dimensional space. By grouping input samples into vectors and using a vector quantizer, a lower bit rate and higher performance can be achieved. However, the codebook size and the computational complexity increase exponentially as the rate increases for a given vector size. Full-search VQ such as entropy-constrained VQ (ECVQ) enjoys small quantization distortion. However, it has long compression time, and may not be well suited for real time signal compression systems. Tree-structured VQ (TSVQ) although can significantly reduce the compression time, has the disadvantage that the storage size required for the VQ is usually very large and cannot be controlled during the design process. Therefore, it may not be convenient to use TSVQ for the applications where the storage size is a major concern. A structured VQ scheme which can achieve very low encoding and storage complexity is MSVQ [12]. This appealing property of MSVQ motivated us to use MSVQ in the quantization stage. The basic idea of multistage vector quantization is to divide the encoding task into successive stages, where the first stage performs a relatively crude quantization of the input vector. Then a second-stage quantizer operates on the error vector between the original and the quantized first-stage output. The quantized error vector then provides a second approximation to the original input vector thereby leading to a refined or more accurate representation of the input. A third stage quantizer may then be used to quantize the second-stage error to provide a further refinement and so on. In this paper, we have implemented two-stage vector quantizer. The input vector is quantized by the initial or first stage vector quantizer denoted by VQ1 whose code book is $C_1 = C_{10}, C_{11} \ldots C_1 (N_{1-1})$ with size N_1. The quantized approximation ˆx1 is then subtracted from x producing the error vector. This error vector is then applied to a second vector quantizer VQ2 whose code book is $C_2 = C_{20}, C_{21} \ldots C_2 (N_{2-1})$ with size N2 yielding the quantized output [10].

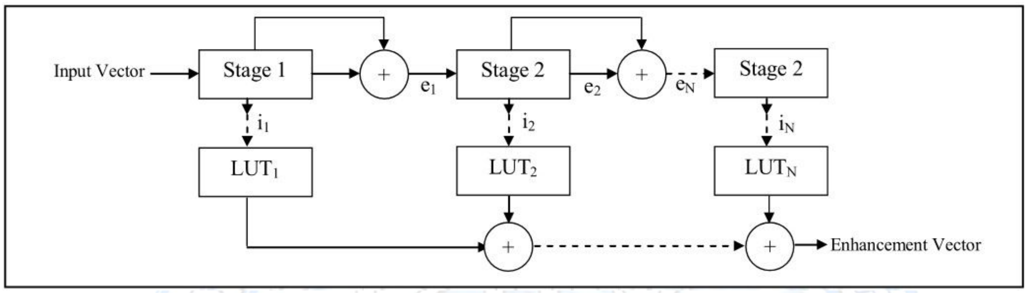

Figure (2): Multistage Vector Quantization System

The encoder transmits a pair of indices specifying the selected codeword for each stage and the task of the decoder is to perform two table lookups to generate and then sum the two code words. In fact, the overall codeword or index is the concatenation of code words or indices chosen from each of two codebooks. Thus, the equivalent product codebook can be generated from the Cartesian product $C_1 \times C_2$. Compared to the full-search VQ with the product codebook C, the two stage VQ can reduce the complexity from $N = N_1 \times N_2$ to $N_1 + N_2$. The multistage vector quantization system for 'N' stages is shown in Fig. 2. In the figure, 'X' represents the input vector, LUT stands for lookup table and i_1 , i_2 , etc represent indices from different stages. The overall index is the concatenation of indices chosen from each of the two codebooks. From the Fig.2, it is evident that the input vector is given only to the first stage, whereas the input to the successive stages is the error vectors from the previous stage [11].

Types of Thresolding and Their Selection Rules

Thresholding is one of the most commonly used processing tools in multiwavelet signal processing. It is widely used in noise reduction, signal and image compression or recognition [12]. In addition, Thresolding is non-linear operation performed on the multiwavelet coefficients of noisy signal [18]. This can be done by comparing the absolute value of the empirical multiwavelet coefficients with a value called Threshold Value (Thv). It is clear that if the multiwavelet coefficient is equal to or less than the threshold value, then one can not separate the signal from the noise. In this case, a good estimation for that multiwavelet coefficient is zero. In the case of an empirical multiwavelet coefficient is greater than the threshold value, then a natural estimation for this multiwavelet coefficient is empirical multiwavelet coefficient itself. This idea is called thresolding, which optimizes the mean squared error of the image. Thresolding process has different types as will be show below. The choice of thresolding method depends on the application. Thresolding operations are applied to the coefficients of the multiwavelet transforms.

Hard-Thresholding

Hard Thresholding is also called "kill/keep" strategy or "gating". If the signal or a coefficient value is below a present value it is set to zero, that is:

$$\hat{A}_K^j = T_h(G_K^j, Thv) = \begin{cases} G_K^j & |G_K^j| > Thv \\ 0 & |G_K^j| \le Thv \end{cases} \quad \dots (16)$$

Where THv is the threshold value or the gate value the graphical representation of the hard threshold is shown in figure (3). Note that the graph is nonlinear and discontinuous at x=Thv [11].

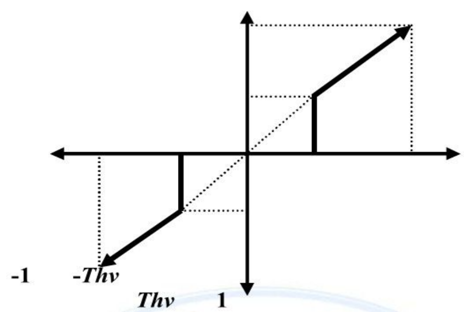

Figure (3): Hard Thresolding

Soft-Thresholding

Soft thresholding is an alternative scheme of hard Thresolding and can be stated as [12]:

$$\hat{A}_K^j = T_h(G_K^j, Thv) = \begin{cases} \text{Sign}(G_K^j)\times|G_K^j|-Thv & |G_K^j|>Thv \\ & |G_K^j|\le Thv \end{cases} \qquad \ldots (17)$$

Hard thresolding can be described as the usual process of setting to zero the wavelet coefficients whose absolute values are less than or equal to the threshold value Thv. soft thresolding is an extension of hard thresolding, firstly setting to zero the wavelet coefficients whose absolute values are less than or equal to the threshold value Thv, then shrinking the non-zero coefficients towards zero by threshold value Thv. Figure (4) shows this type of Thresolding [10

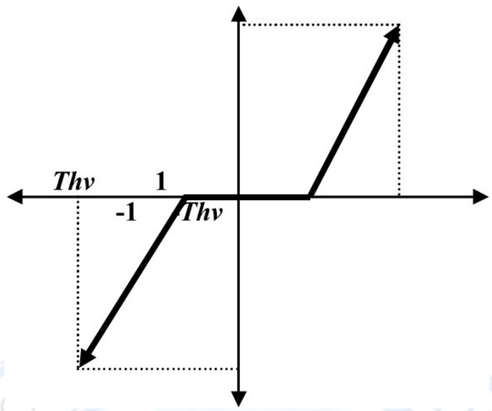

Figure (4): Soft thresolding

proposed method of denoising using (Hybrid Techniques) Mixed-Thresolding

In this section we suggest to use two theory of Thresolding to appear the image in best denoising through using soft threshold and hard threshold too by obtain this from Eqs.(8) and Eqs.(9). This theory can apply by using the soft threshold in the LL and LH part and hard threshold in HL and HH part this theory can named as mixed threshold that can be explain in the figure.4

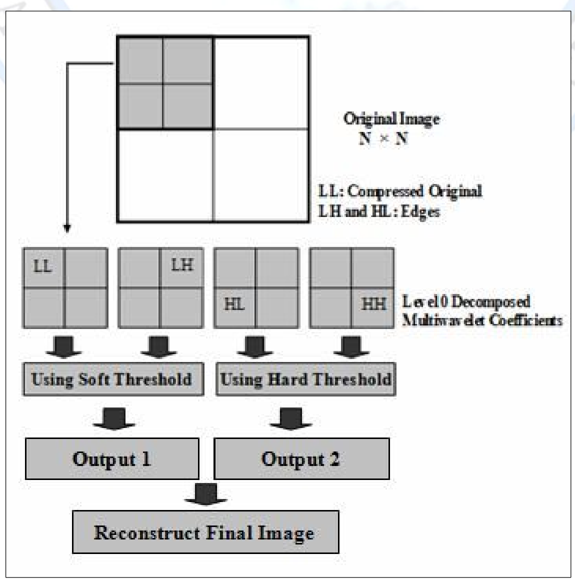

Figure (5): flow diagram of the proposed Multiwavelet thresolding method

General Example for Computing DMWT for decomposition

1. ***Coefficient Shuffling***, which is applied to the DMWT N×N matrix four basic sub bands individually. For each sub bands, coefficient shuffling shuffles the columns first then shuffles the rows.

2. ***Column Reconstruction***,

I. Transpose the coefficient shuffled N×N matrix.

II. Apply shuffling by arranging the row pairs 1, 2 and 3, 4⋯ (N/2)-1, (N/2) of the coefficient shuffled N×N matrix transpose to be the row pairs 1, 2 and 5, 6⋯ N-3, N-2 of the resulting matrix and arranging the row pairs (N/2)+1, (N/2)+2, and (N/2)+3, (N/2)+4⋯ N-1, N of the coefficient shuffled N×N matrix transpose to be the row pairs 3, 4 and 7, 8⋯ N-1, N of the resulting matrix.

III. Multiply an N×N reconstruction matrix (N×N transformation matrix transpose) with the resulting N×N shuffled matrix from II.

3. ***Postprocessing***, a critical-sampled scheme of Postprocessing can be computed as follows:

I. 1^{st} order approximation Postprocessing: can be computed by applying these equations:

Odd-row= [[same odd-row] - (0.11086198) [next even-row] - (0.11086198) [previous even-row]] / (0.373615) ... *(18)*

Even-row= [same even-row] / (√2 -1) ... *(19)*

to the odd- and even-rows of the column reconstructed N×N matrix respectively.

II. 2^{nd} order approximation Postprocessing: can be computed by applying these equations:

Odd-row= [[same odd-row] - (3/8√8) [next even-row] - (3/8√2) [previous even-row]]/ (10/8√2) ... *(20)*

Even-row= [same even-row] ... *(21)*

to the odd- and even-rows of the column reconstructed N×N matrix respectively.

4. ***Row*** **and** ***Column reconstruction***

I. Transpose the Postprocessed N×N resultant matrix.

II. Apply shuffling by arranging the row pairs 1, 2 and 3, 4··· (N/2)-1, N/2 of the N×N matrix Postprocessed resultant matrix transpose to be the row pairs 1, 2 and 5, 6··· N-3, N-2 of the resulting matrix and arranging the row pairs (N/2) +1, (N/2) +2, and (N/2) +3, (N/2)+4··· N-1, N of the N×N matrix Postprocessed resultant matrix transpose to be the row pairs 3, 4 and 7, 8··· N-1, N of the resulting matrix.

III. Apply the ***Multistage Vector Quantization*** for row preprocessing to the input 2-D matrix, *X*, using repeated row preprocessing and c= c10, c11 ...c1 (N1−1)

$$X = \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} \\ x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,0} & x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,0} & x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix} \xrightarrow[\textbf{rows}\ x\,=]{\textbf{preprocess}} \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} \\ cx_{0,0} & cx_{0,1} & cx_{0,2} & cx_{0,3} \\ x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} \\ cx_{1,0} & cx_{1,1} & cx_{1,2} & cx_{1,3} \\ x_{2,0} & x_{2,1} & x_{2,2} & x_{2,3} \\ cx_{2,0} & cx_{2,1} & cx_{2,2} & cx_{2,3} \\ x_{3,0} & x_{3,1} & x_{3,2} & x_{3,3} \\ cx_{3,0} & cx_{3,1} & cx_{3,2} & cx_{3,3} \end{bmatrix} \quad \text{... (22)}$$

1. Apply row transformation

I. Let [z]= [W]× [x]

II. Permute [z],

$$z = \begin{bmatrix} z_{0,0} & z_{0,1} & z_{0,2} & z_{0,3} \\ z_{1,0} & z_{1,1} & z_{1,2} & z_{1,3} \\ z_{2,0} & z_{2,1} & z_{2,2} & z_{2,3} \\ z_{3,0} & z_{3,1} & z_{3,2} & z_{3,3} \\ z_{4,0} & z_{4,1} & z_{4,2} & z_{4,3} \\ z_{5,0} & z_{5,1} & z_{5,2} & z_{5,3} \\ z_{6,0} & z_{6,1} & z_{6,2} & z_{6,3} \\ z_{7,0} & z_{7,1} & z_{7,2} & z_{7,3} \end{bmatrix} \xrightarrow{\textbf{permute}} p = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & p_{0,3} \\ p_{1,0} & p_{1,1} & p_{1,2} & p_{1,3} \\ p_{4,0} & p_{4,1} & p_{4,2} & p_{4,3} \\ p_{5,0} & p_{5,1} & p_{5,2} & p_{5,3} \\ p_{2,0} & p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,0} & p_{3,1} & p_{3,2} & p_{3,3} \\ p_{6,0} & p_{6,1} & p_{6,2} & p_{6,3} \\ p_{7,0} & p_{7,1} & p_{7,2} & p_{7,3} \end{bmatrix}, \quad \dots (23)$$

1. Apply column transformation

I. Transpose p matrix.

$$p^t = \begin{bmatrix} p_{0,0} & p_{1,0} & p_{4,0} & p_{5,0} & p_{2,0} & p_{3,0} & p_{6,0} & p_{7,0} \\ p_{0,1} & p_{1,1} & p_{4,1} & p_{5,1} & p_{2,1} & p_{3,1} & p_{6,1} & p_{7,1} \\ p_{0,2} & p_{1,2} & p_{4,2} & p_{5,2} & p_{2,2} & p_{3,2} & p_{6,2} & p_{7,2} \\ p_{0,3} & p_{1,3} & p_{4,3} & p_{5,3} & p_{2,3} & p_{3,3} & p_{6,3} & p_{7,3} \end{bmatrix} \quad \dots (24)$$

II. preprocess $[p]^t$ to get $[P]$ matrix

$$P = \begin{bmatrix} P_{0,0} & P_{1,0} & P_{4,0} & P_{5,0} & P_{2,0} & P_{3,0} & P_{6,0} & P_{7,0} \\ \alpha P_{0,0} & \alpha P_{1,0} & \alpha P_{4,0} & \alpha P_{5,0} & \alpha P_{2,0} & \alpha P_{3,0} & \alpha P_{6,0} & \alpha P_{7,0} \\ P_{0,1} & P_{1,1} & P_{4,1} & P_{5,1} & P_{2,1} & P_{3,1} & P_{6,1} & P_{7,1} \\ \alpha P_{0,1} & \alpha P_{1,1} & \alpha P_{4,1} & \alpha P_{5,1} & \alpha P_{2,1} & \alpha P_{3,1} & \alpha P_{6,1} & \alpha P_{7,1} \\ P_{0,2} & P_{1,2} & P_{4,2} & P_{5,2} & P_{2,2} & P_{3,2} & P_{6,2} & P_{7,2} \\ \alpha P_{0,2} & \alpha P_{1,2} & \alpha P_{4,2} & \alpha P_{5,2} & \alpha P_{2,2} & \alpha P_{3,2} & \alpha P_{6,2} & \alpha P_{7,2} \\ P_{0,3} & P_{1,3} & P_{4,3} & P_{5,3} & P_{2,3} & P_{3,3} & P_{6,3} & P_{7,3} \\ \alpha P_{0,3} & \alpha P_{1,3} & \alpha P_{4,3} & \alpha P_{5,3} & \alpha P_{2,3} & \alpha P_{3,3} & \alpha P_{6,3} & \alpha P_{7,} \end{bmatrix} \quad \dots (25)$$

III. let $[b] = [W] \times [P]$

IV. Permute $[b]$ to get $[B]$matrix which is 8×8 matrix.

V.

$$b = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,4} & b_{0,5} & b_{0,6} & b_{0,7} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} & b_{1,6} & b_{1,7} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} & b_{2,6} & b_{2,7} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} & b_{3,6} & b_{3,7} \\ b_{4,0} & b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & b_{4,6} & b_{4,7} \\ b_{5,0} & b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} & b_{5,6} & b_{5,7} \\ b_{6,0} & b_{6,1} & b_{6,2} & b_{6,3} & b_{6,4} & b_{6,5} & b_{6,6} & b_{6,7} \\ b_{7,0} & b_{7,1} & b_{7,2} & b_{7,3} & b_{7,4} & b_{7,5} & b_{7,6} & b_{7,7} \end{bmatrix} \quad \ldots (26)$$

$$\Big\downarrow \textbf{Permute}$$

$$B = \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} & B_{0,4} & B_{0,5} & B_{0,6} & B_{0,7} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} & B_{1,4} & B_{1,5} & B_{1,6} & B_{1,7} \\ B_{4,0} & B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & B_{4,5} & B_{4,6} & B_{4,7} \\ B_{5,0} & B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} & B_{5,5} & B_{5,6} & B_{5,7} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} & B_{2,4} & B_{2,5} & B_{2,6} & B_{2,7} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & B_{3,3} & B_{3,6} & B_{3,7} \\ B_{6,0} & B_{6,1} & B_{6,2} & B_{6,3} & B_{6,4} & B_{6,5} & B_{6,6} & B_{6,7} \\ B_{7,0} & B_{7,1} & B_{7,2} & B_{7,3} & B_{7,4} & B_{7,5} & B_{7,6} & B_{7,7} \end{bmatrix} \quad \ldots (27)$$

3. The final **DMWT** matrix $[Y]$ results from the following steps,

I. Transpose $[B]$ matrix to get $[y]$ matrix.

II. Apply coefficients permutation to each of the four basic sub bands of matrix $[y]$.

5. ***Post processing***, a critical-sampled scheme of Post processing can be done by the same process of step 3 above which results in the N×N original reconstructed 2-D signal matrix.

Denoising process for multiwavelet (MS-QIDMWT) and Hybrid threshold

If the noised image

$$I(i,j)=X(i,j)+n(i,j) \quad i,j=1,2,\ldots,N \qquad \ldots(28)$$

Where $n(i, j)$ is white Gaussian noise whose mean value is zero, σ is its variance, and $X(i,j)$ the original signal. The problem of de-noising can be thought as how to recover $X(i, j)$ from $I(i,j)$. Transform the formula (13) with multiwavelet, formula (14) is obtained

$$WI\,(i,j)= Wx(i,j)+ Wn\,(i,j) \qquad \ldots(29)$$

It is known from multi-wavelet transformation that, the multi-wavelet transformation of Gaussian noise is also Gaussian distributed, there are components at different scales, but energy distributes evenly in high frequency area, and the specific signal of the image has projecting section in every high frequency components. So image denoising can be performed in high frequency area of multi-wavelet transformation.

System Flowcharting of Thresholding Algorithm by Using (MSQIDMWT)

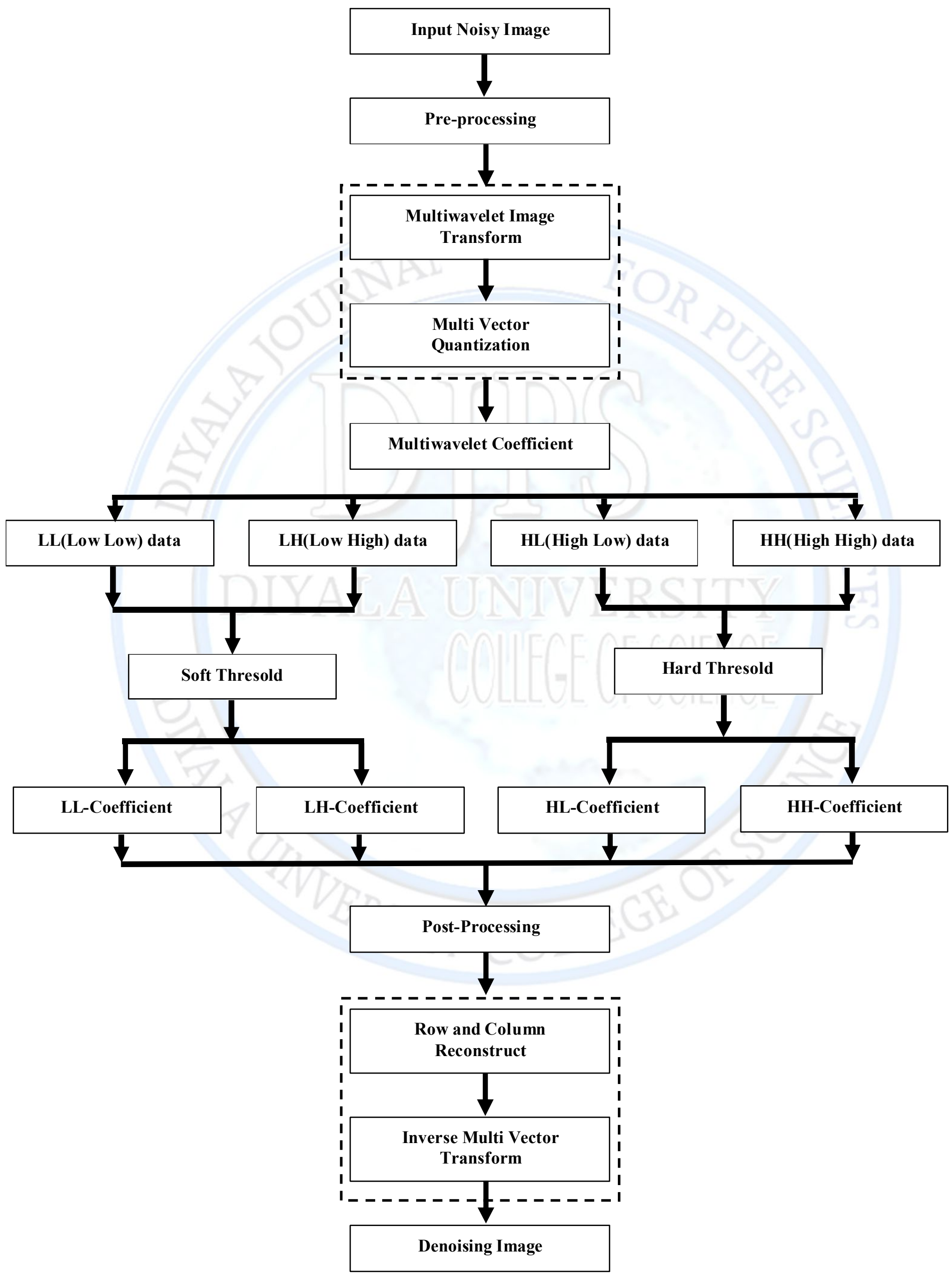

Figure (6): Flowchart of proposed System for multiwavelet hybrid thresholding method

Thresholding Algorithm by Using (MS-QIDMWT)

In this section, a new hard thresholding role is proposed to eliminate the errors which appear in the threshold images. Figure (3.8) shows the block diagram of the method. The algorithm of the proposed method is as follows: (Note that: this algorithm can be used either for hard or soft thresholding)

Step1. Obtain the multiwavelet transform (MS-QIDMWT) coefficients of the observed noisy image (G_K^j).

Step2. Compute the threshold value (*Thv*) by equation or any other one of the threshold selection rules.

Step3. Select the threshold type according the part of coefficients for LL and LH select (soft-threshold) and for HL, HH select (Hard-threshold). Since the work of this research depends on the hybrid techniques mixed between soft and hard thresholding type and filter the multiwavelet transform coefficients (G_K^j).don't change if the absolute value of the coefficients is greater than the threshold value (*Thv*), else replace it with a zero value.

Step4. Inverse the Multiwavelet transform threshold coefficients ($\hat{A}_k^j$) to get the denied (reconstructed) image.

Results and Discussion

A comparison of image denoising using multiwavelet transform with that other images figure (5) shows the different images and compares the result of denoising images by using multiwavelet transform in Gaussian Noise.

Experiment 1

In this experiment result a general comparison of image Denoising using Multiwavelet transform with the other images by using the proposed method of multiwavelet and Hybrid techniques of thresolding.



(a) **(b)** **(c)**



(d) **(e)** **(f)**

Figure (5): Result of applying multiwavelet on denoising depended on Hybrid threshold in images Where the Gaussian noise with mean=0 and σ =20.

In this experiment table (1) shows the different measurement (RMSE, SNR and PSNR) and compares the result of denoising images by using Multiwavelet transform one by using traditional thresholding (Hard in table (1)) and (Soft threshold in table (2)) additional to table(3) shows the different measurement (RMSE, SNR and PSNR) according to the proposed method (Hybrid techniques) .

Table (1): RMSE, SNR, PSNR Objective measurement of images with different Noisy images Gaussian Noise and Denoising Images depended on using hard threshold

	Noisy Image			Desisting using Hard-threshold		
IMAGES	**PSNR**	**SNR**	**RMSE**	**PSNR**	**SNR**	**RMSE**
Tiger	**141.12**	**10.01**	**0.17**	**21.8036**	**20.0056**	**14.5615**
Watch	**187.92**	**9.99**	**0.003**	**20.4561**	**20.1885**	**15.9608**

Table (2): RMSE, SNR, PSNR Objective measurement of images with different Noisy images Gaussian Noise and Denoising Images depended on using soft threshold

	Noisy Image 1			Desisting using Soft-threshold 1		
IMAGES	**PSNR**	**SNR**	**RMSE**	**PSNR**	**SNR**	**RMSE**
Tiger	**141.12**	**I00I**	**0.17**	**21.8036**	**21.0544**	**20.0422**
Watch	**187.92**	**9.99**	**0.003**	**20.4561**	**20.7622**	**21.6092**

Table (3): RMSE, SNR, PSNR Objective measurement of images with different Noisy images Gaussian Noise and Denoising Images depended on using Hybrid thresholding

	Noisy Image 1			Denoising Image 1		
IMAGES	**PSNR**	**SNR**	**RMSE**	**PSNR**	**SNR**	**RMSE**
Tiger	**141.12**	**10.01**	**0.17**	**20.43**	**22.54**	**24.89**
Watch	**187.92**	**9.99**	**0.003**	**20.01**	**24.91**	**25.89**

Experiment 2

In this experiment2 table (4) shows the different measurement (RMSE, SNR and PSNR) and compares the result of denoising images by using Multiwavelet transform one by using traditional thresholding and (Soft threshold in table) additional to shows the different measurement according to the proposed method (Hybrid techniques) [13].

Table (4): Difference Objective measurement RMSE, SNR, PSNR of Denoising Images using hard threshold, Soft threshold and Hybrid techniques

	Desisting using Hard-threshold			Desisting using Soft-threshold			Desisting using 1 Hybrid		
IMAGES	**PSNR**	**SNR**	**RMSE**	**PSNR**	**SNR**	**RMSE**	**PSNR**	**SNR**	**RMSE**
Tiger	21.8036	20.0056	14.5615	21.8036	21.0544	20.0422	20.43	22.54	24.89
Watch	20.4561	20.1885	15.9608	20.4561	20.7622	21.6092	20.01	24.91	25.89

Conclusion

In this paper, technique of multiwavelet transform play an important role in image denoising applications due to its multiresolution analysis that makes the statistics of many natural images be simplified when they are decomposed. It takes the idea of wavelet transform with different two-dimensional filters and with pre/post processing mechanisms. There are two important parameters in multiwavelet thresholding denoising algorithm namely the threshold value (Thv) and the multiwavelet basis. For the first parameter, it is found that the optimal threshold value can be computed. Choosing the multiwavelet basis is not an easy task. GHM, with repeated row preprocessing and matrix approximation preprocessing for different images are employed. Among these basis functions, it is seen that is the performance improves when using CL filter with repeated row preprocessing. The level of decomposition of multiwavelet transform and the thresholding scheme (soft and hard) are two important parameters in multiwavelet thresholding denoising algorithm. Image denoising using the proposed algorithm take the advantage of averaging and translation-invariant in noise elimination. For example, the result of multiwavelet (soft and hard) thresolding basis functions gave an RMSE value of (25.89) but traditional threshold gives (22.28) using hard threshold and (21.66) using soft threshold.

References

1. Michale B. Martin, *"Applications of multiwavelets to Image Compression",* M.Sc. Thesis in Electrical Engineering, Virginia polytechnic Institute and State University(Virginia Tec),Blacksburg,june,2008.
2. Vasily Strela, Peter Niels Heter,Gilbert Strang, Pankaj Topiwala, and Christopher Heil*," The Application of Multiwavelet Filterbank to Image Processing"*, IEE TRANSACTIONS ON IMAGE PROCESSING, VOL.8, NO.4,April 2009.
3. Michael B.Martin and Amy E.Bell, Member, IEE, *"New Image Compression Techniques Using Multiwavelet and Multiwavelet Packets"* IEE TRANSACTIONS ON IMAGE PROCESSING, VOL.10, NO.4 APRIL 2007.
4. G. Y. Chen and T. D. Bui, *"Multi-Wavelet De-Noising Using Neighboring Coefficients,"* IEEE Signal Processing Letters, VOL.10, NO.7, PP.211-214, 2003.
5. L. Sendur and I.W. *Selesnick 2002 Bivariate Shrinkage Functions For Wavelet-Based Denoising Exploiting Interscale Dependency* IEEE Trans Signal Processing.
6. Lin K Z, Li D P and Hua K Q 2008 *Operator Description of Image Wavelet Denoising* Journal of Harbin University of Science and Technology. 5 8 -12.
7. S Q Zhang, X H Xu, J T Lv, X Y Xang and N He OF *An Improved Approach to Image Denoising Based on Multi- Wavelet and Threshold*, International Symposium on Instrumentation Science and Technology, Journal of Physics.
8. Wilsou, R., *Multiresolution Image Modeling, Electronic and Comm*. Journal, PP, 90-96, April 1997.
9. Selesnick, I. W., *"Interpolation Multiwavelet Bases and The Sampling Theorem*, IEEE Transe. On SP, Vol. 47, PP. 1615-1621, 1999.
10. Shen, L. X., Tan, H. H., and Tham, J. Y., *Some Properties of Symmetric- Antisymmetric Orthonormal Multiwavelets*, SP EDICS on multirate processing and wavelets, May 4, 1999.

11. Bronstein, M., and Bornstein, A., Signal and Image Denoising Using Constrained Optimization, concluding report for the seminar (advanced topics in computer vision), April-July, 2001.
12. Goswami, J. C., and Chan, A. K., *Fundamentals of Wavelet Theory*, Algorithms, Applications, John Willy and Sons, 1999.
13. Adil abdulwahhab Ghidan, "Denoising *and Noise Identification Techniques Based on using Multiwavelet Transform and Neural Networks*", M.Sc. Thesis in University of Technology, Iraq, Baghdad, 2005.