TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE

ELECTRICAL ENGINEERING | ELEKTROTECHNIKA

2-E/2015

VOLODYMYR FEDAK, ADRIAN NAKONECHNY*

ADAPTIVE WAVELET THRESHOLDING FOR IMAGE DENOISING USING SURE MINIMIZATION AND CLUSTERING OF WAVELET COEFFICIENTS

ZASTOSOWANIE ADAPTACYJNYCH PROGÓW DO REDUKCJI SZUMÓW OBRAZÓW ZA POMOCĄ MINIMIZACJI SURE I KLASTEROWANIA WSPÓŁCZYNNIKÓW FALKOWYCH

Abstract

Images and video are often coded using block-based discrete cosine transform (DCT) or discrete wavelet transform (DWT) which cause a great deal of visual distortions. In this paper, an extension of the intra-scale dependencies of wavelet coefficients is proposed to improve denoising performance. This method incorporates information on neighbouring wavelet coefficients that are inside of manually created clusters. Extensive experimental results are given to demonstrate the strength of the proposed method.

Keywords: *artifacts*, *denoising*, *wavelet transformation*

Streszczenie

Obrazy i nagrania wideo są często kodowane z użyciem blokowej dyskretnej transformacji kosinusowej (DCT) lub dyskretnej transformacji falkowej (DWT), które powodują znaczne zakłócenia wizualne. W niniejszej pracy proponuje się rozszerzenie zależności między współczynnikami falkowymi dotyczącymi skali w celu zmniejszenia zaszumienia sygnału zakodowanego. Zaproponowana metoda zakłada wykorzystanie informacji o sąsiadujących współczynnikach falkowych, które znajdują się wewnątrz manualnie utworzonego klastra. W artykule zaprezentowano obszerne wyniki doświadczalne w celu wykazania jakości proponowanej metody.

Słowa kluczowe: *artefakty*, *odszumianie*, *transformacja falkowa*

DOI: 10.4467/2353737XCT.15.096.3928

* Ph.D. student Volodymyr Fedak, Prof. Adrian Y. Nakonechny, Computer System And Automatics, Lviv Polytechnic National University.

1. Artifacts in images

All the time, we need to obtain, process, and deliver information. This information is not just limited to text files or simple messages; nevertheless, various visual pieces of information can be transmitted including image and video files. However, transmission channels have limited bandwidth and storage devices hold limited capacity. Digital video is broadcasted and stored in an encoded form; therefore, it requires less information (bits) than the original. At low bit-rates, the coarse quantization exploited during compression results in visually annoying coding artifacts [1].

Compression artifacts are a particular class of data errors that are usually the consequence of quantization in loss-prone data compression. These distortions can be classified into the following types:

Blocking artifacts. This type of image distortion is the most visible degradation of all artifacts. This effect is caused by all block-based coding techniques. It is a well-known fact that all compression techniques divide the image into small blocks and then compress them separately. Due to the coarse quantization, the correlation among blocks is lost, and horizontal and vertical borders appear.

Ringing artifacts. The ringing effect is caused by the quantization or truncation of high frequency coefficients and can also come from improper image restoration operations. Ringing artifacts are visible for all compression techniques especially when the image is transformed into the frequency domain. Moreover, it appears as distortion along sharp edges in the image. This artifact occurs very often when the DWT encoder is used. Furthermore, it may be observed after the image has been de-coded using a frequency coder.

Blur effect. Blurring is another artifact resulting from the absence of high frequencies in low bit rate video. It appears around sharp edges, and all image details become blurred. This effect is very similar to the ringing artifact, and sometimes it is hard to distinguish between them.

Different techniques may be used to reduce the most annoying artifacts and all of these techniques can be divided by filtering domain (spatial, frequency). Different authors provide versatile methods of image quality improvements and sometimes, the most challenging task is to choose the necessary technique. Spatial algorithms modify image pixel values. These approaches are usually used together with edge detection algorithms to prevent the blurring effect, many classical image denoising methods are based on a local average. The restored value at a pixel is obtained as an average of its neighboring pixels. The most classical algorithm is Gaussian filtering. In this case, the restored value is obtained as a weighted average where the weight of each pixel depends on the distance to a restored one – this low pass filter tends to blur the image. The neighborhood filters avoid the blurring effect by restricting the average to pixels having a similar grey level value. The idea is that grey level values inside a homogeneous region slightly fluctuate while pixels belonging to different regions have a larger grey level difference. The neighborhood filter takes an average of the values of pixels which are simultaneously close in their grey level values and spatial distance. The most encouraging results can be obtained using the NLM approach [2]. Efficiency of this algorithm is proven in many different areas and this algorithm tries to take advantage of the redundancy and self-similarity of the image. The NLM algorithm estimates the value of x as an average of the values of all the pixels. The probability that one pixel is similar to the other is determined by looking at the difference in the luminance value and the difference in position between two pixels in the neighbourhood filters.

Another direction of image restoration is using wavelet based techniques.

The multi-resolution analysis performed by the WT has been shown to be a powerful tool in order to achieve good denoising. In the wavelet domain, the noise is uniformly spread throughout the coefficients, while most of the image information is concentrated in the few largest coefficients (sparsity of the wavelet representation).

The most straightforward way of distinguishing information from noise in the wavelet domain consists of thresholding the wavelet coefficients [3–5]. Using a soft-thresholding filter is the most popular strategy and has been theoretically justified by Donoho and Johnstone [6]. They propose a three steps denoising algorithm:

1) the computation of the forward WT,
2) the filtering of the wavelet coefficients,
3) the computation of the IWT of the result obtained.

They use the discrete wavelet transform (DWT) and the soft-thresholding filter. Because it is not made any explicit hypothesis on the noise-free image, it results in a non-parametric method.

In this article, an efficient algorithm based on adaptive thresholding and wavelet coefficients dependencies for image denoising is presented. The proposed algorithm can effectively reduce noise in static images. The experimental results show that the proposed approach significantly outperforms the BiShrink, Neigh Shrink and Block Shrink approaches. Evaluations have been performed on different images and with different noise levels.

2. Existing approaches for image denoising

In modern digital systems and video broadcast chains, image compression is applied to reduce bandwidth or storage size. Post-processing of the decoded image sequence is an acceptable technique to achieve a better perceived picture quality [10]. Furthermore, modern consumer vision products like televisions and PCs use image enhancement and restoration techniques to improve the objective and subjective picture quality. All postprocessing algorithms and methods can be divided into the following types [1]:

– spatial filtering,
– filtering in the frequency/wavelet domain,
– temporal filtering,
– hybrid algorithms (mainly combines spatial and frequency filtering).

Many approaches have been proposed in the literature aiming at the alleviation of artifacts in the images. As a great number of algorithms have been developed in recent times, it would be rational to overview these approaches which threshold the wavelet detail coefficients for two-dimensional (2-D) signals. An overview of spatial techniques has been conducted in our previous work [7].

Frequency algorithms transform images to frequency domains and modify DCT (discrete cosine transform) or DWT (discrete wavelet transform) coefficients.

The effect of averaging the spatially closest pixels can also be achieved in the Fourier or wavelet domain. The average of the spatially closest pixels is then equivalent to the cancellation of the high frequencies. As the analogous spatial filter, this cancellation leads to the blurring of the image and a Gibbs effect. The optimal filter in the Fourier domain is the Wiener filter – this does not cancel the high frequencies but attenuates all of them.

In the wavelet domain, the noise is uniformly spread throughout the coefficients, while most of the image information is concentrated in the few largest coefficients (sparsity of the wavelet representation).

Consequently, regarding the three steps denoising algorithm, there are two tools to be chosen: the WT (wavelet transform) and the filter. In [8] the UDWT (undecimated discrete wavelet transform) is used, in [9] the DTCWT (dual tree complex wavelet transforms), and in [10] the DWT.

From the first category, we can mention: the hard-thresholding filter that minimizes the min-max estimation error and the efficient SURE-based inter-scales point-wise thresholding filter [10], which minimizes the mean square error (MSE). To the second category belong filters obtained by minimizing a Bayesian risk under a cost function, typically a delta cost function (MAP estimation [11]) or the minimum mean squared error [8]. The denoising algorithms proposed in [10] exploit the inter-scale dependence of wavelet coefficients. The method proposed in [8] also takes into account the intra-scale dependence of wavelet coefficients. The statistical distribution of the wavelet coefficients changes from scale to scale. The coefficients of the WT have a heavy tailed distribution.

2.1. Image filtering using wavelet thresholding

The wavelet denoising methods filter each coefficient from the detail sub-bands with a threshold function to obtain modified coefficients. The denoised estimated by inverse wavelet transform of the modified coefficients. Here, the threshold plays an important role in the denoising process. There are two thresholding functions frequently used. The soft--threshold function (also called the shrinkage function):

$$soft(d,\lambda) = sign(d) * [|d| - \lambda]_+ \tag{1}$$

takes the argument and shrinks it towards zero by the threshold *T*. The other alternative is the hard-thresholding function:

$$hard(d,\lambda) = \begin{cases} d, \text{if } |d \geq \lambda| \\ 0, \text{if } |d < \lambda| \end{cases} \tag{2}$$

which keeps the input if this input is larger than the threshold; otherwise, it is set to zero. The wavelet thresholding procedure removes noise by only thresholding the wavelet coefficients of the detail sub-bands, while keeping the low resolution coefficients unaltered.

Hard thresholding keeps existing coefficients whereas soft thresholding shrinks the coefficients above the threshold in absolute value.

More frequently, it's started to use adaptive threshold. Our proposed denoising method also uses such a type of thresholding (Fig. 2).

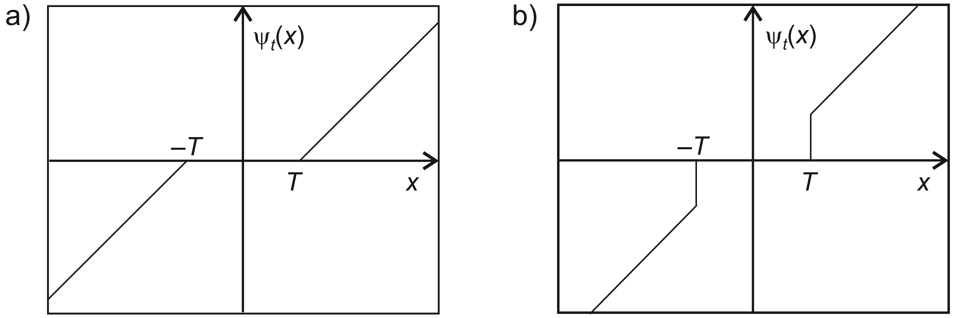

Fig. 1. Thresholding functions: a) soft threshold, b) hard threshold

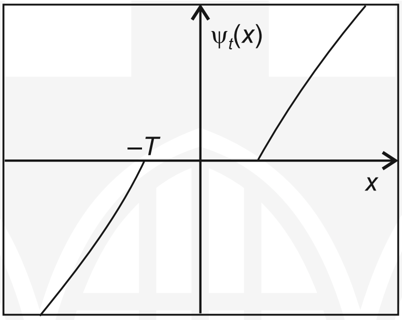

Fig. 2. Adaptive thresholding function

The adaptive thresholding doesn't modify wavelet coefficients that have high magnitudes (this is a drawback of soft thresholding) and doesn't have disruptions of wavelet coefficients close to T (this is a drawback of hard thresholding).

2.2. Wavelet dependencies and shrinkages types

In general, four types of wavelet coefficient dependencies can be considered. This is illustrated in Fig. 3:

c) Intra-scale and intra-band dependencies,
d) Inter-scale and inter-band dependencies,
e) Inter-band and intra-scale dependencies,
f) Inter-scale and intra-band dependencies.

The most frequently used de-nosing methods use only two types of dependencies: intra-scale and intra-band, and inter-scale and intra-band dependencies. The wavelet coefficients that represent the image also have large magnitudes at these scales, locations and orientations (d). However, the signs and relative magnitudes of these coefficients will depend on the exact shape, location and orientation of the structure they represent. The inter--scale dependencies indicate that if a parent coefficient is large, then its child coefficient is also large. However, currently there are no strict correlations between parent/child

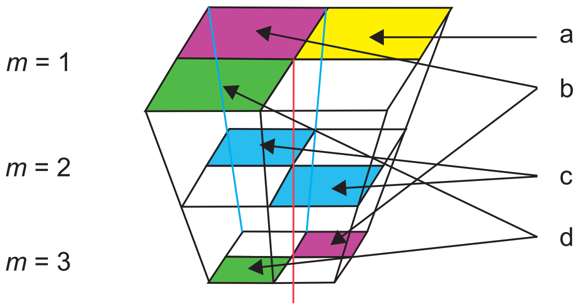

Fig. 3. Wavelet coefficient’s dependencies

wavelet coefficients [11]. It’s possible to indicate such a type of dependencies but only between two consecutive scales.

The method that uses inter-scale dependencies is called bivariate shrinkage [11]. The bivariate shrinkage function can be interpreted as follows:

$$w_1 = \frac{\left(\sqrt{y_1^2 + y_2^2} - \frac{\sqrt{3\sigma_n^2}}{\sigma}\right)}{\sqrt{y_1^2 + y_2^2}} y_1 \tag{3}$$

where y_1, y_2 – noisy coefficients in the scale 1 and 2. σ is defined as marginal variance. This estimator requires prior knowledge of the noise variance σ_n. In order to estimate noise variance, a robust median estimator is used from the finest scale wavelet coefficients [12].

$$\sigma_n = \frac{median\left(|w_s|\right)}{0.6745} \tag{4}$$

And marginal variance is defined as:

$$\sigma = \sqrt{(\sigma_y^2 - \sigma_n^2} \tag{5}$$

where σ_y is defined as:

$$\sigma_y^2 = \frac{1}{M} \sqrt{\sum_{y_i \in N(k)} y_i^2} \tag{6}$$

Another method that uses intra-scale dependencies is called Neigh Shrink [13]. The shrunken wavelet coefficient according to the neigh shrink is given by this formula:

$$\overline{w}_{i,j} = w_{i,j}\beta_{i,j} \tag{7}$$

where $w_{i,j}$ – noisy wavelet coefficient; $\overline{w}_{i,j}$ – denoised wavelet coefficient; the shrinkage factor $\beta_{i,j}$ can be defined as:

$$\beta_{i,j} = (1 - T_{UNI}^2 / S_{i,j}^2)_+ \tag{8}$$

There, the + sign at the end of the formula means to keep the positive value while set it to zero when it is negative and T_{UNI} is the universal threshold, which is defined as:

$$T = \sqrt{2\sigma \ln(N)} \tag{9}$$

And $S_{i,j}^2$ is a summation of the wavelet coefficients in the neighbouring window centered at the wavelet coefficient to be shrunk (Fig. 4):

$$S_{i,j} = \sum_{\in B_{i,j}} w_{k,l}^2 \tag{10}$$

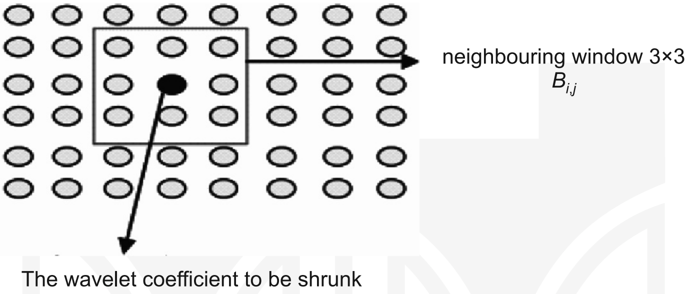

Fig. 4. Wavelet coefficients of neighbouring window

Another method that uses the SURE optimization technique (that minimizes the mean square error) is a block shrink. Suppose $w = \{w_i,\ i = 1, 2, \ldots, d\}$ and $S_B^2 = \sum_{\in B_{i,j}} w_{k,l}^2$. If S_B^2 is less than or equal to the threshold λ, then within the b block, the wavelet coefficient w_i is set to zero. Otherwise, wavelet coefficient is shrunk according to (7). The optimal threshold λ and block size L are derived by minimizing Stein's unbiased risk estimate (SURE) [14]:

$$(\lambda^s, L^s) = \arg_{\lambda,L} SURE(w,\lambda,L) \tag{11}$$

where:

$$SURE(w,\lambda,L) = \sum_{b}^{m} SURE(w_b,\lambda,L) \tag{12}$$

and

$$\begin{cases} SURE(w_b,\lambda,L) = L + \dfrac{\lambda^2 - 2\lambda(L-2)}{S_b^2} & \text{if} \quad (S_b^2 > \lambda) \\ S_b^2 - 2L \quad \text{if} \quad (S_b^2 > \lambda) \end{cases} \tag{13}$$

$m = d/L$ is the number of blocks.

3. Image denoising using SURE minimization and clustering of wavelet coefficients

The proposed method of wavelet shrinkage uses adaptive wavelet thresholding. In order to find a better threshold, the proposed approach uses SURE minimization and intra-scale and intra-band dependencies of wavelet coefficients. The general flow chart of the proposed wavelet shrinkage algorithm for noise reduction is depicted in Fig. 5. The main idea is that the signal is transformed using DWT to the wavelet domain and thereafter LH, HL, and HH sub bands are processed. The wavelet coefficients processing consists of the following steps:

- clustering; clusters are created from wavelet coefficients (all details are described in section 3.1),
- determining the threshold parameter for each cluster type (details are in section 3.2),
- thresholding (details in section 3.3).

After this processing, wavelet coefficients are again transformed to the original format.

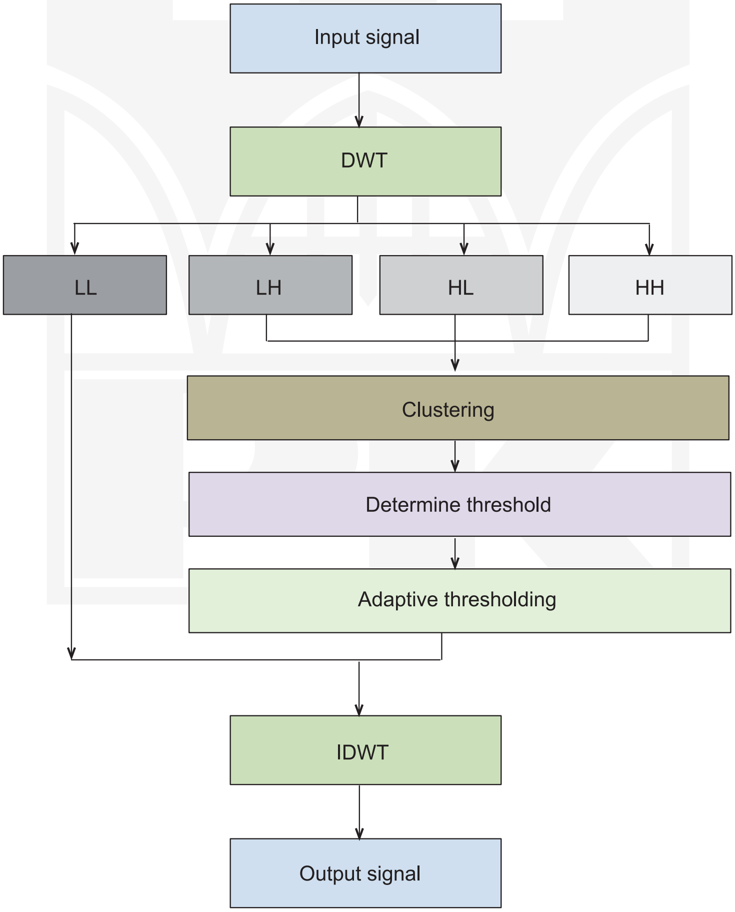

Fig. 5. Flow chart of the proposed wavelet shrinkage algorithm for noise reduction

3.1. Wavelet coefficients clustering

It has been determined that better results of image denoising can be obtained after clustering of the wavelet coefficients (coefficients need to be clustered in the LH, HL, HH). Better results can be received when cluster satisfies the following requirements:

- the absolute average value of wavelet coefficients shouldn't be much greater than the wavelet coefficient to be shrunk,
- denoising provides better results when a cluster contains more than 6–12 wavelet coefficients,
- if wavelet coefficients in the cluster have a greater value, the obtained results should be better.

Thus, the proposed method uses the following clusters (Fig. 6).

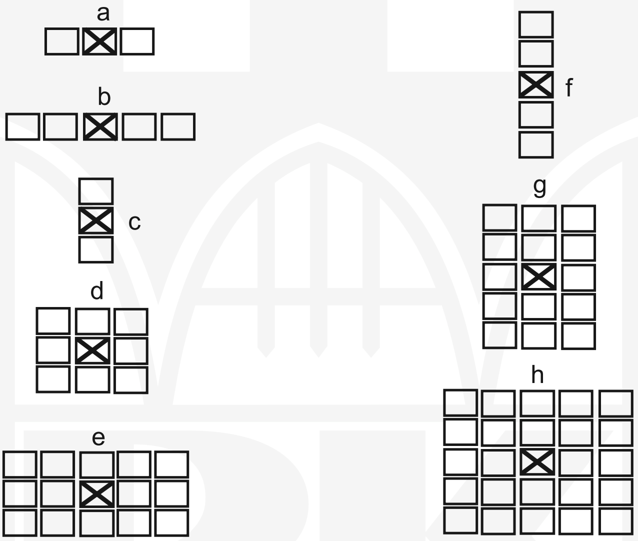

Fig. 6. Wavelet clusters that are formed from 5×5 blocks

Practically greater blocks are used: 6x6 and 7×7 blocks; therefore, the amount of different cluster types are greater than is displayed on Fig. 6.

Let w_o be the estimator of the unknown nosiness coefficient. Each cluster is created for the purpose of thresholding each noisy coefficient $w_{i,j}$:

$$A_{x,y} = \arg\max_{B_{i,j}} \left(\frac{1}{i \cdot j} \sum_{A_{x,y}} w_{i,j}^2 \right) \tag{14}$$

where $A_{x,y}$ cluster that is found from wavelet coefficients that have maximum value in the block $B_{i,j}$, $w_{i,j}$ – wavelet coefficients that belong to block $B_{i,j}$.

Additional weights are added to the clusters that have more than 4 × 3 = 12 wavelet coefficients:

$$Acl = 1.3 \cdot \frac{1}{i \cdot j} \sum_{A_{x,y}} w_{i,j}^2 \quad (\text{if} \quad x \cdot y > 12) \tag{15}$$

where:

Acl – a square average of wavelet coefficient inside cluster $A_{x,y}$,
1.3 – determined experimentally (using a database with images).

As is mentioned, additional weight is added to the cluster that contains wavelet coefficients with a similar value to the shrunken wavelet coefficient:

$$Acl = 15 \cdot \frac{1}{i \cdot j} \sum_{A_{x,y}} w_{i,j}^2 \quad (\text{if} \quad abs(w_o^2 - w_{i,j}^2) < 1.4 w_o^2) \tag{16}$$

3.2. Determine the better threshold parameter

The next step is to choose an appropriate threshold value. Different wavelet coefficients are shrunken differently with different threshold values that are found for different cluster types. The optimal λ and *L* of every sub band should be data-driven and minimize the mean squared error (MSE) or Steins risk of the corresponding sub band and for specified cluster type (Stein proved that MSE can be found unbiasedly for a given noise level. For these purposes, a robust median estimator can be used).

In the proposed algorithm, the optimal threshold λ for each cluster type are derived by minimizing Stein's unbiased risk estimate:

$$SURE(w_o, \lambda, L) = N_s + \sum_n \| g_n(w_n) \|_2^2 + 2 \sum_2 \frac{\delta g_n}{w_n} \tag{17}$$

where:

L – cluster type (Fig. 6),
λ – threshold,
N_s – a noisy coefficient.

$$\| g_n(w_n) \|_2^2 = \begin{cases} \frac{\lambda^4}{S_n^4} w_n^2 & (\lambda < S_n) \\ w_n^2 & (otherwise) \end{cases} \tag{18}$$

where S_n is the summation of wavelet coefficients in the cluster *L*; and $\frac{\delta g_n}{w_n}$

$$\frac{\delta g_n}{w_n} = \begin{cases} -\lambda^2 \frac{S_n^2 - 2w_n^2}{S_n^4} & (\lambda < S_n) \\ -1 \quad (otherwise) \end{cases} \tag{19}$$

3.3. Thresholding

Thresholding is conducted in a similar manner as to how it's performed in the Neigh shrink method [13]. The wavelet coefficients are shrunken according to the Neigh shrink using this formula:

$$\overline{w}_{i,j} = w_{i,j}\beta_{i,j} \tag{20}$$

where:

$w_{i,j}$ – noisy wavelet coefficient,

$\overline{w}_{i,j}$ – denoised wavelet coefficient,

the shrinkage factor $\beta_{i,j}$ for specified cluster types is defined as:

$$\beta_{i,j} = (1 - T_{UNI}^2 / S_{i,j}^2)_+ \tag{21}$$

and $S_{i,j}^2$ is a summation of the wavelet coefficients in the cluster centered at the wavelet coefficient to be shrunk:

$$S_{i,j} = \sum_{\in B_{i,j}} w_{k,l}^2 \tag{22}$$

4. Results

In this section, an objective analysis of the proposed wavelet shrinkage algorithm with an adaptive thresholding is evaluated. We have experimented with various noisy images and report the results for the three 512×512 standard test images: Lena, Barbara and Mandrill (Fig. 6). The DWT is used with a symlet wavelet with eight vanishing moments with four scales. They are contaminated with Gaussian random noise with standard deviations of 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100. In practice, the noise standard deviation is unknown and during experiments, it's initially estimated using (4). Our results are measured by the PSNR in decibels (dB) and MSE.

Our method was compared with BiShrink that uses inter-scale intra-band and intra-scale intra-band wavelet coefficients dependencies. In addition, we compared our method

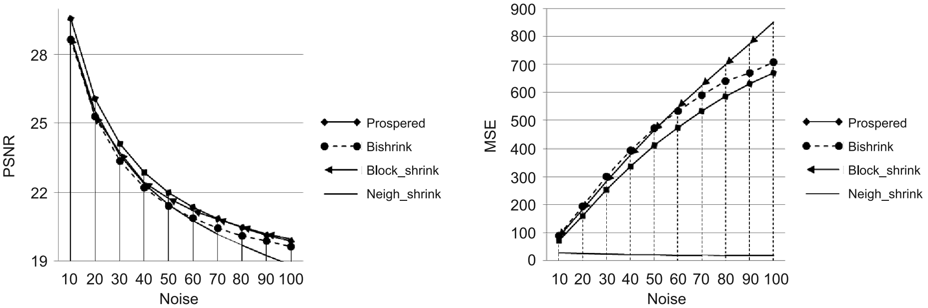

Fig. 7. Comparison of the PSNR and MSE for the Barbara image on different noise levels

with the Neigh shrink, according to paper [15], this method produces the most promising results. In comparison to other methods, the Block shrink (only this method is based on SURE) also produced quite good results. The comparison charts for the Mandrill image is depicted on Fig. 7.

Extensive results are described in the Table 1 (‘*’ indicates the best result among all four denoising methods).

Table 1

Comparison of PSNR and MSE for different wavelet methods

Image	Sigma	Proposed		BiShrink		Block_shrink		Neigh_shrink	
		PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE
Mandrill	10	**32.842***	**33.796***	32.140	39.729	31.594	45.048	32.714	34.809
	20	**29.000***	**81.847***	28.253	97.222	27.695	110.556	28.758	86.555
	30	**26.903***	**132.657***	26.159	157.455	25.919	166.399	26.498	145.648
	40	**25.519***	**182.483***	24.786	215.991	24.798	215.399	24.933	208.827
	50	**24.514***	**229.980***	23.828	269.321	23.995	259.183	23.759	273.622
	60	**23.740***	**274.808***	23.179	312.725	23.392	297.763	22.830	338.938
	70	**23.119***	**317.085***	22.691	349.942	22.914	332.384	22.062	404.434
	80	**22.633***	**354.600***	22.296	383.272	22.517	364.228	21.396	471.456
	90	**22.174***	**394.161***	21.942	415.816	22.166	394.884	20.802	540.652
	100	21.813	428.316	21.629	446.896	**21.828***	**426.83***	20.261	612.364
Barbara	10	**32.842***	**33.796***	32.140	39.729	31.594	45.047	32.713	34.809
	20	**29.001***	**81.847***	28.253	97.222	27.695	110.556	28.757	86.555
	30	**26.904***	**132.657***	26.159	157.455	25.919	166.399	26.497	145.647
	40	**25.519***	**182.483***	24.786	215.991	24.798	215.398	24.932	208.827
	50	**24.514***	**229.980***	23.828	269.320	23.994	259.182	23.759	273.621
	60	**23.741***	**274.808***	23.179	312.724	23.392	297.762	22.829	338.937
	70	**23.12***	**317.085***	22.691	349.942	22.914	332.383	22.062	404.434
	80	**22.633***	**354.600***	22.296	383.271	22.517	364.228	21.396	471.455
	90	**22.174***	**394.161***	21.942	415.816	22.166	394.884	20.801	540.651
	100	21.813	428.316	21.628	446.896	**21.828***	**426.83***	20.260	612.363

Lena	10	**34.610***	**22.493***	34.322	24.033	33.821	26.97	34.533	22.894
	20	**31.380***	**47.328***	31.126	50.171	30.720	55.085	31.083	50.666
	30	**29.524***	**72.559***	29.370	75.171	29.152	79.046	28.895	83.849
	40	**28.219***	**97.984***	28.106	100.562	28.042	102.050	27.274	121.783
	50	**27.231***	**123.025***	27.141	125.597	27.166	124.851	25.989	163.714
	60	**26.46*0**	**146.908***	26.407	148.704	26.431	147.883	24.924	209.258
	70	**25.807***	**170.764***	25.779	171.840	25.802	170.922	24.004	258.622
	80	25.238	194.666	25.114	200.276	**25.255***	**193.902***	23.171	313.249
	90	24.733	218.680	24.730	218.809	**24.777***	**216.442***	22.430	371.585
	100	24.29	242.313	24.235	245.215	**24.346***	**239.009***	21.750	434.541

As described in Table 1, the proposed method provides better results from among all mentioned denoising methods. Results are especially better for images with high texture details. In comparison to Neigh Shrink, the difference is more than 3 db when the amount of noise is high. However, for images with fewer textures block shrink also provides quite good results and when the amount of noise is high, these results are even slightly better than in the proposed method.

5. Conclusions

The presented method demonstrated that the adaptive wavelet thresholding method for image denoising using SURE minimization and clustering of wavelets provides significant improvement in comparison to the existing image denoising methods. In order to show the effectiveness of the new algorithm, several examples are presented and compared with effective techniques in the literature. The objective analysis demonstrated that the proposed method significantly outperforms the BiShrink, Neigh shrink and block shrink approaches. Evaluations have been performed on different images and with different noise levels. Results are especially better for images with high texture details. However, for images with fewer textures, block shrink also provides quite good results and when the amount of noise is high, these results are even slightly better than in the proposed method.

This article demonstrated that the clustering of the wavelet coefficient might be beneficial and also provides a new direction for future research. This technique can especially be used altogether with other techniques which explore inter-scale dependencies or can even be combined with other image denoising methods.

References

[1] Fedak V.I., Nakonechny A.Y., *Artifacts suppression in images and video. Non-Local Means as algorithm for reducing image and video distortions*, Hluboka nad Vltavou, Czech Republic 2009.

[2] Buades A., *Image and film denoising bynon-local means*, Ph.D. Thesis in Journal of Visual Commincation and Image Representation, March 2004, Vol. 1, No. 1, pp. 2–4, Article No. VC970378.

[3] Luisier F., Blu T., Unser M., *A New SURE Approach to Image Denoising: Inter-Scale Orthonormal Wavelet Thresholding*, IEEE Transactions on Image Processing, 2007, Vol. 16, No. 3, pp. 593–606.

[4] Zhou Z.-F., Shui P.-L., *Contourlet-based image denoising algorithm using directional windows*, Electronic Letters, 2007, Vol. 43, No. 2, pp. 92–93.

[5] Olhede S.C., *Hyperanalytic denoising*, IEEE Transactions on Image Processing, 2007, Vol. 16, No. 6, pp. 1522–1537.

[6] Donoho D.L., Johnstone I.M., *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, 1994, Vol. 81, No. 3, pp. 425–455.

[7] Fedak V., Nakonechny A., *Spatio-temporal algorithm for coding artifacts reduction in highly compressed video*, Technical Transactions Automatic Control, 2013.

[8] Pizurica A., Philips W., *Estimating the probability of the presence of a signal of interest in multiresolution single and multiband image denoising*, IEEE Transactions on Image Processing, 2006, Vol. 15, No. 3, pp. 654–665.

[9] Shui P.-L., *Image Denoising Algorithm via Doubly Local Wiener Filtering With Directional Windows in Wavelet Domain*, IEEE Signal Processing Letters, 2005, Vol. 12, No. 6, pp. 681–684.

[10] Luisier F., Blu T., Unser M., *A New SURE Approach to Image Denoising: Inter Scale Orthonormal Wavelet Thresholding*, IEEE Transactions on Image Processing, 2007, Vol. 16, No. 3, pp. 593–606.

[11] Achim A., Kuruoglu E.E., *Image Denoising Using Bivariate -Stable Distributions in the Complex Wavelet Domain*, IEEE Signal Processing Letters, 2005, Vol. 12, No. 1, pp. 17–20.

[12] Donoho D.L., Johnstone I.M., *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, 1994, Vol. 81, No. 3, pp. 425–455.

[13] Chen G.Y., Bui T.D., *Multi-wavelet Denoising using Neighboring Coefficients*, IEEE Signal Processing Letters, 2003, Vol. 10, No. 7, pp. 211–214.

[14] Stein C., *Estimation of the mean of a multivariate normal distribution*, Ann. Statist., 1981, Vol. 9, pp. 1135–1151.

[15] Rohit Sihag, Rakesh Sharma, Varun Setia Wavelet, *Thresholding for Image Denoising*, International Conference on VLSI, Communication & Instrumentation (ICVCI), 2011.