COMBINING THE DISCRETE WAVELET TRANSFORMS AND NEAREST NEIGHBOUR FILTERS FOR IMAGE RESTORATION

Karen O. Egiazarian, Jaakko T. Astola, Samvel M. Atourian, and David Z. Gevorkian

Signal Processing Laboratory, Tampere University of Technology,
P. O. Box 553, FIN-33101, Tampere, Finland. E-mail: karen, jta, sam, gevdav@cs.tut.fi

ABSTRACT

A combination filter structure involving wavelet transform based denoising methods and K−nearest neighbor (K-NN) type operations is proposed and studied. Performance analysis of this filter shows its high efficiency in suppressing mixed white Gaussian and impulsive noises. At the same time the proposed filter possess moderate computational complexity.

1. INTRODUCTION

The most widely used methods of image restoration in the presence of impulsive type noises are based on rankings of the pixels in neighborhood according to brightness. Spatial nonlinear filters (see, [1, 5, 6, 10]), preserve the image sharpness and remove efficiently certain kinds of impulsive noise, like "shot" noise (when individual pixels corrupted or missing from image).

Among such techniques note the K nearest neighbor (K−NN) filter introduced by Davis and Rosenfeld [5]. This filter performs well for both cases of additive and multiplicative noise [8]. However, an application of the K-NN filter was restricted due to its main drawback - very high computational time. A modification of the K−NN filter with much lower (of about 20 times less) complexity has been suggested in [9]. In [2] another modification of the K−NN filter has been proposed which for even slightly corrupted images gives practically the same output as the K−NN filter does, but is even simpler in implementation compared to the modification of [9]. An efficient bit-serial implementation of this modified K−NN filter is developed in [2].

On the other hand, for removing additive white Gaussian noise from signals and images, wavelet transform based denoising methods have been proven to perform excellent [4]. They work pretty well also in several applications where the error is neither white nor Gaussian [11]. These applications are noise reduction (de-noising) of synthetic aperture radar (SAR) signals, medical and geophysical signals, as well as removing blocking artifacts in images of JPEG decoded signals [11]. However, the Donoho's method (wavelet-based) for noise reduction is not working in the case when image is corrupted by even a small percentage of impulses. It comes natural to incorporate positive sides of wavelet denoising methods and K−NN type filtering operations in a unified filtering structure for removing mixed type noises.

In this paper we propose a novel filter structure based on the superposition of a modified K−NN filter and the discrete wavelet transform based filter. The combined filter performs well in suppressing mixed impulsive and Gaussian noises. At the same time the proposed filter can be implemented with moderate computational complexity.

2. THE FILTER STRUCTURE

In general, the filter structure we study in this paper is formed as the superposition of an impulse removal filter G and a wavelet transform based filter F:

$$Y = F(G(X)), \qquad (1)$$

where $X = \{x(i,j), i = 1,2,\dots,I, j = 1,2,\dots,J\}$, $x(i,j) \in \{0,1,...,R-1\}$, $R = 2^r$, is an observation of an image $U = \{u(i,j), i = 1,2,\dots,I, j = 1,2,\dots,J\}$ corrupted by a mixed white Gaussian and an impulsive noise, and $Y = \{y(i,j), i = 1,2,\dots,I, j = 1,2,\dots,J\}$, is the output of the filter which tends to estimate the image U.

2.1. The wavelet denoising method

As the filter F in (1) consider the noise reduction (denoising) method by nonlinear thresholding in the wavelet domain proposed by Donoho and Johnstone [4]. This method consists of the following three steps (we consider here the 1-D case; in the case of 2-D separable transform the image is first transformed row-by-row and then column-by-column):

1. Transform the noisy data into an orthogonal domain,

$$\mathbf{z} = W\mathbf{x} \qquad (2)$$

2. Apply thresholding to the resulting coefficients, which will result in suppression of the coefficients of lower energy, using so-called *hard thresholding* or *soft thresholding*:

$$\hat{\mathbf{z}} = T_h(\mathbf{z}, t) = \begin{cases} \mathbf{z}, & |\mathbf{z}| \geq t, \\ 0, & |\mathbf{z}| < t \end{cases} \qquad (3)$$

$$\hat{\mathbf{z}} = T_s(\mathbf{z}, t) = \begin{cases} sgn(\mathbf{z})(|\mathbf{z}| - t), & |\mathbf{z}| \geq t, \\ 0, & |\mathbf{z}| < t. \end{cases} \qquad (4)$$

3. Transform back to the original domain, performing the inverse transform:

$$\mathbf{y} = W^{-1}\hat{\mathbf{z}}. \qquad (5)$$

However, applying this classical wavelet de-noising scheme in practice, one may end up with some artifacts near singularities (the pseudo-Gibbs phenomena in the neighbourhood of discontinuities). One way to overcome this problem is to use undecimated (or shift-invariant) wavelet transforms [4], which can be done according to the following simple strategy: wavelet de-noising is applied for all circular shifts of a signal, each of the particular result of de-noising is unshift, and, finally, the average of all these results is obtained [4].

Donoho has shown that soft thresholding is the l_2 optimal, and the resulting error is within a logarithmic factor of the ideal risk (which is a measure of performance of an ideal scheme). Hard thresholding, however, does not guarantee the smoothness property but have better l_2 performance instead, especially for shift-invariant wavelet transform-based schemes. As an alternative to these two different thresholding strategies we will use middle case, by applying semi-hard "staircase" thresholding given by

$$\hat{\mathbf{z}} = \frac{1}{m}\sum_{i=1}^{m} T_h(\mathbf{z}, t_i), \qquad (6)$$

where $T_h(Y, t_i)$ are hard threshold operators defined by (3).

A proper selection of thresholds t_i is quite important. There are many different ways to do that. Donoho gives a "universal threshold" [4]. As a value for t the Donoho's threshold is $t = \sigma\sqrt{2\log(n)}$, where n is the length of the signal $\mathbf{x}$, $\mathbf{x} = \mathbf{u} + \sigma\mathbf{n}$, $\mathbf{u}$ is the corresponding noiseless signal, and $\mathbf{n}$ is an additive Gaussian white noise. In general, t_i can be selected as $t_i = c_i\sigma$, where c_i are some constants.

2.2. The impulse removal filter

Specific to the filter structure of (1) is that the impulse removal filter G should not destroy frequencies in the image which is essential for the second stage, in the wavelet de-noising[1]. This means that while removing impulses the filter should not change image pixels where no impulse has caused. Such a filter G would not blur edges which cannot be reconstructed at the second stage. Thus, it is desirable to apply such an impulse removal filter at the first stage that process image pixels with different filtering actions depending on the probability of appearing impulses in the neighbourhood of the given pixel. This kind a strategy is used in *state-conditioned filters* in contrast to the case of conventional rank-order based filters [1, 7] where every pixel is processed uniformly.

In state-conditioned techniques the filtering procedure is conditioned on the current state of the algorithm [1, 6]. The output of the filter is defined as:

$$y(i,j) = \sum_{k=1}^{K} \alpha_{k,s(i,j)} u_k(i,j), \; i = 1, ..., I, \; j = 1, ..., J \quad (7)$$

where $u_k(i,j)$, $k = 1, , , ., K$, are different estimates of the image pixel $u(i,j)$, $s(i,j) \in \{1, ..., S\}$ is a state variable that classifies the current pixel to one of the S categories, and $\alpha_{k,s(i,j)}$ are the scalar coefficients corresponding to each category. For example, the ROM filter proposed in [1] uses $u_1(i,j) = x(i,j)$ and $u_2(i,j) = ROM(i,j)$ as the estimates,

[1] Otherwise, a problem of finding G which transfers the image in a form more suitable to wavelet denoising rather than the given observation after removing impulses could be considered. However this is a very complicated task.

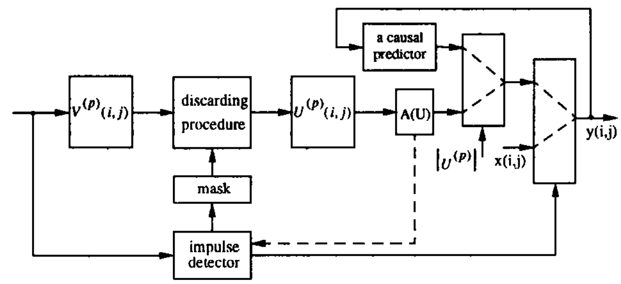

Figure 1: The general structure of the impulse removal filter

where $ROM(i,j)$ is the rank-ordered mean (the average of the fourth and fifth order statistics within the (3×3) window excluding the central pixel).

The minimum-maximum exclusive mean (MMEM) filter recently introduced in [6] uses four different estimates: $u_1(i,j) = x(i,j)$, $u_2(i,j) = mean(U^{(3)}(i,j))$, $u_3(i,j) = mean(U^{(5)}(i,j))$, and $u_4(i,j) = mean(Y(i-1, j\pm1), Y(i-1, j), Y(i, j-1))$, where $U^{(p)}(i,j)$, $p = 3, 5$, is a set of pixel values which is formed by removing some pixels from the $(p \times p)-$window $V^{(p)}(i,j)$ centered at $x(i,j)$ according to a strategy based on computation of minimums and maximums over $V^{(p)}(i,j)$. A comparative study of this filter with other existed filters [6] has demonstrated sufficiently high performance in removing impulsive noises from highly corrupted images. However, its performance becomes relatively worse for images corrupted with small percentage of impulses which is often the case in real applications.

The general structure of the filter which we propose to incorporate into the combination structure of (1) is presented in Figure 1. For every pixel $x(i,j)$, $i = 1, ..., I$, $j = 1, ..., J$, its neighborhood within the $(p \times p)$ window $V^{(p)}(i,j)$ is analysed. An impulse detector is used to find the positions of impulses within $V^{(p)}(i,j)$. Different techniques can be used for this purpose [1, 6]. (Note that in application to restoration of images with missing samples there is no need for impulse detector since positions of missing samples are known.) If the pixel $x(i,j)$ is not recognized as an impulse then no filtering action is applied and the output of the filter is the input pixel $x(i,j)$ itself. Otherwise, the output of the filter is obtained as the result of an operation A applied to the set $U^{(p)}(i,j)$. This set is obtained from the window $V^{(p)}(i,j)$ by discarding some of the pixels according to the mask set obtained by the impulse detector. (Note that assuming positions of impulses are ideally detected (as in the case of missing samples) the set $U^{(p)}(i,j)$ will not contain impulses.) However, if there is no sufficient number of pixels in the set $U^{(p)}(i,j)$ a causal predictor (linear or nonlinear) is used instead of the operation A. The dashed line in Figure 1 indicates that the filtering operation A can be incorporated in the impulse detector.

Thus, the actual filtering action in the proposed structure of the impulse removal filter of Figure 1 is performed in the block A. Below we describe a modified $K-$nearest neighbor type operation which we use as the basic operation of the block A in our experiments. We call the resulting filter G the selective $K-$NN filter.

Let $U^{(p)}(i,j) = \{x(1), x(2), \ldots, x(M)\}$, when the filter's window is located at the pixel $x(i,j)$, $i = 1, ..., I$, $j = 1, ..., J$, and let two integers $K = [\delta M]$ ($M = |U|, 0 < \delta \leq 1$) and $x^c \in \{0, ..., R-1\}$ be given. Some possible choises for x^c are e.g. the center pixel $x(i,j)$ within the window $V^{(p)}$, the median within $V^{(p)}$, the averave of selected samples within $V^{(p)}$, *etc.* Consider the set $D = D(i,j) = \{d(m) = |x(m) - x^c|, m = 1, 2, \ldots, M\}$ and let $d_{(K)}$ be the $K-$th order statistic (the $K-$th smallest value) within D. The result of the modified $K-$NN type operation is defined as:

$$y = A(U^{(p)}) = \left[\sum_{m=1}^{M} f(m)\right]^{-1} \sum_{m=1}^{M} f(m)x(m), \qquad (8)$$

where

$$f(m) = \begin{cases} 1 & \text{if } d(m) \leq d_{(K)} \\ 0 & \text{otherwise} \end{cases}, \quad m = 1, 2, ..., M. \qquad (9)$$

In the partricular case where $U^{(p)}(i,j) = V^{(p)}(i,j)$ (no pixel is discarded) and $\delta = 1$ this becomes simple averaging procedure. In another case where $U^{(p)}(i,j) = V^{(p)}(i,j) \backslash \{x(i,j)\}$ and $x^c = x(i,j)$ the modified $K-$NN operation of (8) performs very similar to the operation of the classical $K-$NN filter, especially in the presence of a noise [2]. However, while the latter one is computationally very complicated, the former one allows a simple bit-serial implementation which we summarize in the next subsection.

2.3. Implementation of the modified $K-$NN type operation

Finding the mask set $f(m)$, $m = 1, 2, ..., M$ is the most complicated part in implementation of the modified $K-$NN type filtering operation. In a straightforward implementation the mask set could be found by finding the $K-$th order statistic in the set D and then using M additional comparisons. This is unefficient, especially because the sliding window designs for computation of order statistics cannot be utilized. We show that a binary-tree search technique resulting in an efficient bit-serial implementation of this filter can be applied. In our implementation we find the mask set f_m without finding the value of the $K-$th order statistic in D. The proposed binary-tree search algorithm for computation of the output of the modified $K-$NN filter over a fixed window can be summarized as follows.

Algorithm 1.

Input. A set $U = \{x(1), x(2), \ldots, x(M)\}$ and an integer x^c, $x(m), x^c \in \{0, 1, \ldots, 2^r - 1\}$.

Output. $y' = \left[\sum_{m=1}^{M} f_m\right]^{-1} \left[\sum_{m=1}^{M} f_m x(m)\right]$, where $f(m)$ are defined by (9).

Computation (pseudocode).

Step 1.

For $m = 1, 2, \ldots, M$ **set** $d(m) = |x(m) - x^c|$.

Step 2.

For $m = 1, 2, \ldots, M$ **set** $f_m = 1$.

Step 3.

Set $T = 2^{r-1}$ /* Note that the numbers of bits in $d(m)$ and $x(m)$ are the same, r */.

Step 4.

For $s = 1, 2, \ldots, r$ **do**

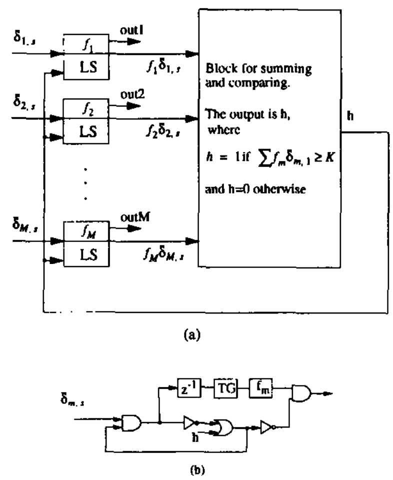

Figure 2: The bit-serial architecture for the modified $K-$NN type operation.

Begin

For $m = 1, 2, \ldots, M$ **set**

$$d_{m,s} = \begin{cases} 1 & \text{if } d(m) < T \\ 0 & \text{otherwise} \end{cases}; \qquad (10)$$

If $\sum_{m=1}^{M} f_m d_{m,s} \geq K$

then set $T = T - \left[2^{r-s-1}\right]$ **and**

For $m = 1, 2, \ldots, M$ **set** $f_m = d_{m,s}$

else set $T = T + \left[2^{r-s-1}\right]$

End

Step 5.

Compute

$$y' = \left[\sum_{m=1}^{M} f_m\right]^{-1} \left[\sum_{m=1}^{M} f_m x(m)\right]$$

Stop.

Algorithm 1 can be efficiently implemented in a bit-serial manner where the sequence $\{d_{m,s}\}$ is obtained by examining the bits of differences $d(m)$ without introducing the parameter T and without performing the comparisons of (10). Let $(\delta_{m,1}\delta_{m,2}...\delta_{m,r})$ be the binary code of $d(m)$, i.e. $d(m) = \sum_{s=1}^{r} \delta_{m,s} 2^{r-s}$.

Proposition. The binary sequence $d_{m,s}$, $m = 1, ..., M$, $s = 1, ..., r$ of Algorithm 1 can be obtained from the binary codes of differences $d(m)$ according to the following rules:

$$d_{m,s} = \begin{cases} 0 & \text{if } \exists q \in \{1, 2, ..., s-1\} \text{ such that} \\ & \sum_{m=1}^{M} f_m^{(q)} d_{m,q} < K \text{ and } d_{m,q} = 0 \\ \bar{\delta}_{m,s} & \text{otherwise} \end{cases}, \qquad (11)$$

where $\bar{\delta}$ is the binary negation of δ, $f_m^{(q)} = \prod_{v \in V_q} d_{m,v}$, $V_q \subset \{1, 2, ..., q-1\}$ is the subset of indices such that $\sum_{m=1}^{M} f_m^{(v)} d_{m,v} \geq K$.

With this proposition Algorithm 1 can be implemented in the bit-serial architecture shown in Figure 2. Bits of the differences are serially entered to the architecture. The values of all the flags are set to high when a new window of differences enters to the circuit Then they are recursively changed according to (11). This is done in logic switches (LS) which are driven by the output h of the block for summing and comparing. A possible realization of logic switches is shown in Fig. 3(b).

Thus the mask set f_m is obtained in $r = \log R$ cycles, where at every cycle only one compare and $M - 1$ adds are implemented. Thus the total complexity of the modified $K-$NN type operation using the proposed implementation is $(r + 2)(M - 1)$ adds and r compares ($2(M - 1)$ adds are needed for summings in (8)). For comparison the histogram-based algorithm of [9] for another modification of the $K-$NN filter requires $2q$ compares, $2\sqrt{M} + 5q - 2$ adds, and $4(q - 1)$ multiplies where q is equal to the difference between the values of the $K-$th and the first order statistics in the filter's window. It may vary between 0 and $R = 2^r$. It should be also noted that the algorithm of [9] is difficult to parallelize and is not suitable for VLSI implementation whereas our method is implemented in a simple bit-serial architecture.

3. PERFORMANCE ANALYSIS AND EXPERIMENTAL RESULTS

We investigated the performance of the proposed filters for restoration of images corrupted by a mixed noise. Extensive simulations have been caried out with a variety of test images and noise models.

Let $X = \{x(i,j), i = 1, 2, \ldots, I, j = 1, 2, \ldots, J\}$, be an observation of an image $U = \{u(i,j), i = 1, 2, \ldots, I,$ $j = 1, 2, \ldots, J\}$ corrupted by mixed noise consisting of i.i.d. zero mean, white Gaussian noise $\sigma n(i,j)$, $n(i,j) \overset{i.i.d}{\sim} \mathcal{N}(0,1)$, with the standard deviation σ and an impulsive noise.

Combination of the proposed impulse removal filter based on modified nearest neighbor type operation with wavelet denoising scheme makes the resulting filter very efficient and robust also for removing mixed Gaussian and impulsive type noises. In our experiments we choose 8-bit 256×256 "Goldhill" as a test image. This image was distorted by mixed Gaussian noise ($\sigma^2 = 100$) and "salt-and-pepper" noise (10%) and the restored image obtained by using our proposed combined filter. Our proposed 3×3 impulse removal filter, MMENN was based on k-nearest neighbor operation using min-max impulse detecting strategy [6]. We use two dimensional translation invariant orthogonal wavelet filter banks (12-taps "Coiflet"), and as the thresholding strategy - semi-hard thresholding. The PSNRs and MAEs for the noisy and restored images are, respectively, PSNR(noisy image)=15.3 dB, PSNR(restored image)=29.9 dB, MAE(noisy image)=19.7, MAE(restored image)=5.9 (see Figure 3).

In Table 1 we show MAEs and PSNRs obtained by applying different filters to image "Goldhill" corrupted by mixed Gaussian (variance 100) and "salt-and-pepper" (4% impulses) noise. We have choosen the following filters: MED, 3×3 median filter, LUM filter (3×3) with parameters $k = 3, l = 5$ [7], k-nearest neighbor filter (KNN) (3×3) [5]. Table 2 and Figure 4 show the results of applying above mentioned filters with wavelet denoising postprocessing.

Table 1: PSNRs and MAEs obtained by different nonlinear filters for corrupted image "Goldhill" (mixed Gaussian (variance 100) and impulsive "salt-and-pepper" (4% impulses) noise)

	MED	KNN	LUM
PSNR	27.14	28.16	27.94
MAE	7.66	6.95	7.01

Table 2: PSNRs and MAEs obtained by different nonlinear filters with wavelet denoising postprocessing for corrupted image "Goldhill" (mixed Gaussian (variance 100) and impulsive "salt-and-pepper" (4% impulses) noise)

	MED	KNN	LUM	MMENN
PSNR	27.43	28.34	28.61	30.49
MAE	7.32	6.74	6.30	5.58

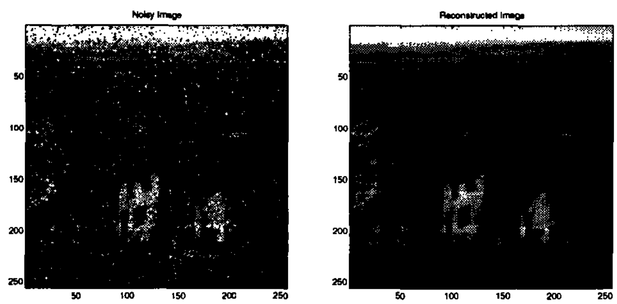

Figure 3: *Left :* Noisy "Goldhill" image (corrupted by Gaussian with variance 100 and "salt-and-pepper" 10 %) *Right :* Restored "Goldhill" image by applying our combined filter

4. REFERENCES

[1] E. Abreu, M. Lightstone, S.K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulsive noise from highly corrupted images," *IEEE Trans. Image Proc.*, 1996, vol. 5, No 6, pp. 1012-1025.

[2] S. Atourian, J. Astola, D. Gevorkian, and K. Egiazarian, "Bit-serial architecture for $K-$nearest neighbor filters," submitted to the Conf. "Computer Science and Information Technologies, (CSIT'97), Yerevan, Armenia, 25-29 Sept. 1997."

[3] M.R. Banham and A.K. Katsaggelos, "Digital Image Restoration", *IEEE Signal Proc. Magazine*, vol. 14, No. 2, pp. 24-41, 1997.

[4] R.R. Coifman and D.L. Donoho, "Translation-Invariant de-noising," in *Wavelets and statistics*, pp. 125-150, Springer-Verlag, 1995.

[5] L.S. Davis and A. Rosenfeld, "Noise cleaning by iterated cleaning," *IEEE Trans. Syst. Man Cybernet.*, vol. SMC-8, 1978, pp. 705-710.

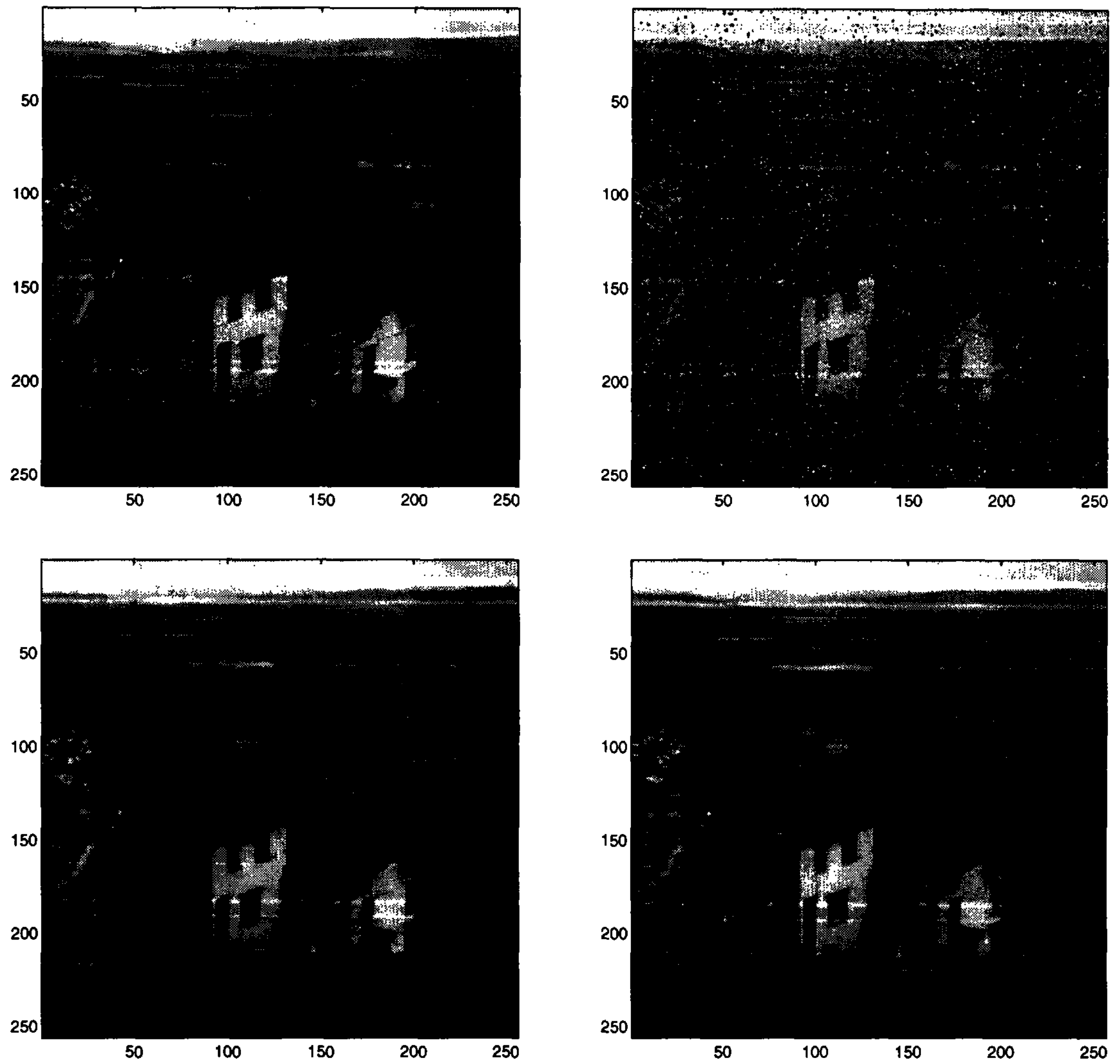

Figure 4: *Left up :* "Goldhill" image. *Right up :* Noisy "Goldhill" image (corrupted by Gaussian with variance 100 and "salt-and-pepper" 4 %) *Left Bottom:* Restored "Goldhill" image by k-nearest neighbor filter with wavelet denoising postprocessing *Right Bottom:* Restored "Goldhill" image by applying our combined filter

[6] W.-Y. Han and j.-Ch. Lin, "Minimum-maximum exclusive mean (MMEM) filter to remove impulse noise from highly corrupted images," *Electronic Letters* vol. 33, no 2, pp. 124-125, Jan. 16, 1997.

[7] R.C. Hardie and C.G. Boncelet, "LUM filters: a class of rank-order-based filters for smoothing and sharpenning," *IEEE Trans. on Signal Processing,* vol. 41, pp. 1061-1075, Mar. 1993.

[8] G.A. Mastin, "Adaptive filters for digital image noise smoothing: An evaluation," *Comput. Vision Graph. Image Process.,* vol. 31, 1985, pp. 103-121.

[9] H.B. Mitchell and Nachum Mashkit, "Noise smoothing by a fast $K-$nearest neighbor algorithm," Signal Processing: *Image Communication,* vol.4, No. 3, June 1992, pp. 227-232.

[10] I. Pitas and A. Venetsanopoulos, *Nonlinear Digital Filters,* Boston: Kluwer Academic Publisher, 1990.

[11] M. Lang, H. Guo, J.E. Odegard, C.S. Burrus, and R.O. Wells, "Nonlinear processing of a shift invariant DWT for noise reduction", *Proc. of SPIE,* vol. 2491, 1995, pp. 640-651.