International Journal of Innovative Computing, Information and Control
Volume **8**, Number **5(A)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3189–3202

A NOVEL DECISION-BASED MEDIAN-TYPE FILTER USING SVM FOR IMAGE DENOISING

Tzu-Chao Lin

Department of Computer Science and Information Engineering
WuFeng University
No. 117, Sec. 2, Chiankuo Rd., Minhsiung, Chiayi 62153, Taiwan
tclin@wfu.edu.tw

Received October 2010; revised February 2011

Abstract. *A novel decision-based filter based on support vector machines (SVMs) that preserves image details and effectively suppresses impulsive noise is proposed. The filter employs an SVM impulse detector to judge whether an input pixel is noisy. If a noisy pixel is detected, a noise-free lower-upper-middle (LUM) filter is triggered to replace it; otherwise, it stays unchanged. To improve the quality of the restored image, an adaptive LUM filter based on scalar quantization (SQ) is employed. The optimal weights of the adaptive LUM filter are obtained using the least mean square (LMS) learning algorithm. Experimental results demonstrate that the proposed scheme outperforms existing decision-based median filters in terms of noise suppression and detail preservation. The proposed filter is also robust against various levels of impulsive noise.*
Keywords: Support vector machine, Least mean square, Impulsive noise, Image restoration

1. **Introduction.** The imperfection of communication channels and image sensors introduces impulsive noise in the form of outliers and bit errors [1-3]. Impulsive noise is a common type of noise that often corrupts digital images. Noise signals are usually strong enough to be sensed by the human eye. Image pre-processing techniques play a key role in image processing applications such as computer vision and pattern recognition. Denoising is thus an essential step before image compression, image retrieval, image copyright protection, edge detection, and object recognition for image processing [32-35]. Image denoising is concerned with not only how to efficiently remove impulsive noise, but also how to preserve image details. Developing an effective image denoising technique has become increasingly important for image processing applications.

A number of approaches have been developed for image denoising. Due to its efficient suppression of impulsive noise, the median filter is a well-known nonlinear filter. However, while suppressing impulsive noise, the median filter sometimes removes fine details. In recent years, variants of the median filter such as weight median (WM) filters, fuzzy-rule-based filters and decision-based filters have been developed in an attempt to improve it [1,4-18]. Satisfactory results have been achieved using these filters. Nevertheless, many WM filters tend to mistakenly alter noise-free pixels. Arakawa et al. proposed a median-type filter controlled by fuzzy rules [4]. An extension of the method, called the partition fuzzy median filter, has also been developed [9]. In fuzzy-based schemes, the optimal weights of the mutually exclusive blocks are realized by training with a reference image. Although good filtering results can be achieved, the filter has a relatively large number of parameters and the generalization capability is poor.

The effectiveness of decision-based filters depends on the effectiveness of the detection process. The switching median (SWM) filter by Sun and Neuvo is based on a threshold

value [16]. The final output switches between the median filter and identity filter. The tri-state median (TSM) filter comprises a median filter, an identity filter and a center weighted median (CWM) filter; it is a special case of the WM filter [6,17,18]. Noise detection is realized by an impulse detector, which takes the outputs from the CWM filter and median filter and compares them with the input pixel value to make a tri-state decision. Recently, a decision-based adaptive low-upper-middle (DALUM) filter was developed [36]. In order to keep the noise-free pixels unchanged, the probability of the local contrast entropy in the filter window is used to judge whether the input pixel is noisy. Generally, the practice of threshold function in decision-based filters is critical in obtaining high noise reduction rates and preserving details. The latest advancement is the adaptive two-pass median (ATM) filter based on support vector machines (SVMs) by Lin and Yu, called the support vector classifier (SVC)-based filter [1]. The SVC-based filter can be regarded as a decision-based filter. It first utilizes SVMs to classify the signal as either noise-corrupted or noise-free and then applies a noise-free reduction filter to remove corrupted pixels. It preserves image details while efficiently suppressing impulsive noise. However, to improve filtering performance, the SVC-based filter requires proper threshold values for a pre-assumed noise density level in second-pass filtering.

In the present study, a novel adaptive SVC-based (ASVC) filter based on SVMs and a least mean square (LMS) learning algorithm that overcomes the drawbacks of existing methods is proposed. The proposed ASVC filter comprises an SVM impulse detector and two low-upper-middle (LUM) smoothers [21,22]. First, the ASVC filter follows the SVM approach; an impulse detection algorithm is used to separate noise-corrupted pixels from noise-free pixels. The accuracy of the SVM impulse detector is satisfactory. Then, only the detected noise-corrupted pixels are filtered; the noise-free pixels are kept unchanged to better preserve image details. The ASVC filter resembles the ATM filter, but its parameters are easily obtained for various noise ratios. The ASVC filter does not require the threshold parameter for LUM filtering. The proposed adaptive LUM filter uses an adjustable weight to best balance the tradeoff between impulsive noise suppression and image detail preservation. The scalar quantizer (SQ) method and a learning approach based on the LMS algorithm are employed to obtain the optimal weight for each block independently [19,20]. With this filtering framework, the proposed ASVC filter significantly outperforms existing median-based filters in terms of noise suppression and detail preservation. The proposed filter is also robust against various impulsive noise ratios.

The rest of this paper is organized as follows. In Section 2, the basic idea of the adaptive LUM filter is introduced. In Section 3, the design of the proposed ASVC filter is presented in detail. In Section 4, the results of some extensive experiments demonstrate that the proposed ASVC filter outperforms existing median-based median filters. Finally, the conclusion is given in Section 5.

2. LUM Filters.

In this section, the algorithms of SVMs for binary classification can be found in [23-30]. Then, LUM filters are introduced.

Let $C = \{(k_1, k_2) | 1 \leq k_1 \leq H, 1 \leq k_2 \leq W\}$ denote the pixel coordinates of the noisy image corrupted by impulsive noise, where H and W are the image height and width, respectively. Let $x(k)$ represent the input pixel value of the noisy image at location $k \in C$. At each location k, the observed filter window $w\{k\}$, whose size is $N = 2n + 1$ (n is a non-negative integer), is defined in terms of the coordinates symmetrically surrounding the input pixel $x(k)$.

$$w\{k\} = \{x_f(k) : f = 1, 2, \ldots, n, n+1, \ldots, N\}, \tag{1}$$

where the input pixel $x(k) = x_{n+1}(k)$ is the center pixel.

Nonlinear low-upper-middle (LUM) smoothers, which are a subclass of LUM filters that take advantage of the computational efficiency of order-statistics-based operators, have been shown to be equivalent to center weighted median (CWM) filters [21,22]. LUM smoothers are defined as:

$$y(k) = MED\{x_{(l)}(k), x(k), x_{(N-l+1)}(k)\}, \tag{2}$$

where MED denotes the median operation, $1 \leq l \leq (N+1)/2$, $x(k)$ is the central sample from the filter window $w\{k\}$, and $x_{(1)}(k) \leq x_{(2)}(k) \leq \ldots \leq x_{(N)}(k)$ is the rank-ordered set of $w\{k\}$. Here, $x_{(l)}(k)$ and $x_{(N-l+1)}(k)$ are the lower-order statistic and upper-order statistic, respectively, and l is the control weight for the smoothing. The amount of smoothing done by LUM smoothers can range from $y(k) = x(k)$ (for $l = 1$) to the median (for $l = (N+1)/2$).

The weight l of non-adaptive LUM filters is a compromise between noise suppression and detail preservation. However, LUM filters uniformly process the whole noisy image, which leads to possible excessive or insufficient smoothing. In the present study, to best balance the tradeoff between impulsive noise suppression and image detail preservation, the weight l of the LUM filter is made adjustable according to the local feature of the filter window.

3. Design of the ASVC Filter.

3.1. **Structure of the ASVC filter.** The framework of the proposed ASVC filter is illustrated in Figure 1. The filter comprises an SVM impulse detector and two LUM filters. The SVM impulse detector is first used to efficiently determine whether the LUM filter or the identity filter should be used. The input pixels are first identified by the SVM impulse detector as either noise-corrupted or noise-free. Then, according to the results of the SVM impulse detector, either the LUM filter is used to remove impulsive noise or the identity filter (no filtering) is used to preserve true pixels. The first LUM filter can remove most of the noise, but some impulsive noise might remain. In order to alleviate this problem, an adaptive LUM filter is used to remove residue impulses with small signal distortion.

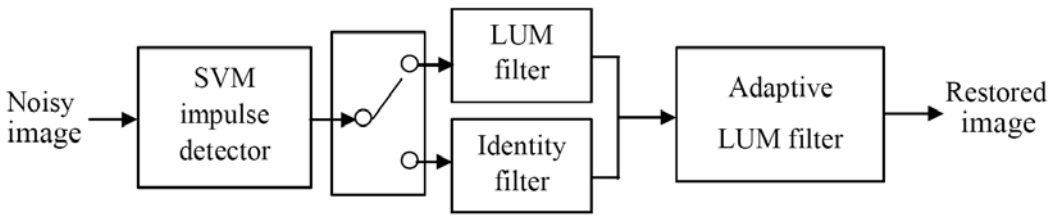


FIGURE 1. Structure of ASVC filter

3.2. **SVM impulse detector.** Decision-based median filters are used to avoid smoothing the image during filtering. They mainly use a detection process to separate noise-free pixels from noise-corrupted pixels. Pixels are left unchanged if they are judged as noise-free. In the present study, an SVM impulse detector based on the SVM approach is proposed to identify noise. This design includes two steps: (1) feature extraction and (2) training of the SVM impulse detector.

3.2.1. *Feature extraction.* Since noise filtering effectiveness heavily depends on the correctness of noise detection, in designing decision-based filters, the SVM impulse detector should be made as precise as possible. Before noise filtering begins, the local features of the filter window $w\{k\}$ must be extracted to identify noisy pixels [1]. The local features in the filter window, such as prominent signals and the possible presence of details and edges, are taken into account. The following three variables are defined to generate feature vector $f\{k\}$ as the input data of the SVM impulse detector.

Definition 3.1. *The variable $c(k)$ denotes the absolute difference between the input $x(k)$ and the median value of $w\{k\}$ as follows* [1,9]*:*

$$c(k) = |x(k) - MED(w\{k\})| .$$

A large $c(k)$ value indicates that the central pixel $x(k)$ stands out among its neighboring pixels; that is, the input $x(k)$ may be corrupted by impulsive noise. Most impulsive noise can be detected by using the variable $c(k)$ as an indicator. However, if only the $c(k)$ value is used to judge whether impulsive noise exists, it would be difficult to fully separate impulsive noise. For example, line components and edges are usually present in an image; therefore, if $x(k)$ is located on a line or an edge, it may be mistakenly interpreted as impulsive noise and removed. To avoid misjudgments, it is necessary to add other observations. Therefore, two extra variables, $l(k)$ and $e(k)$ are used.

Definition 3.2. $c^{w_0}(k) = MED\{x_1(k), \ldots, x_n(k), w_0 \diamondsuit x_{n+1}(k), \ldots, x_N(k)\}$, *where*

$$\begin{aligned} &MED\{x_1(k), \ldots, x_n(k), w_0 \diamondsuit x_{n+1}(k), \ldots, x_N(k)\} \\ &= MED\{x_1(k), \ldots, x_n(k), \underbrace{x_{n+1}(k), \ldots, x_{n+1}(k)}_{w_0 \text{ times}}, \ldots, x_N(k)\}. \end{aligned}$$

here, w_0 denotes a non-negative integer weight, and $w_0 \diamondsuit x_{n+1}(k)$ means that there are w_0 copies of input pixel $x(k) = x_{n+1}(k)$ [11].

Definition 3.3. $l(k) = |x(k) - c^3(k)|$.

Definition 3.4. $e(k) = |x(k) - c^5(k)|$.

If the variable $l(k)$ is used, then a pixel on an edge component in the filter window $w\{k\}$ will not be detected as noise because of its small $l(k)$ value. In addition, if the variable $e(k)$ is used, then a pixel on a line component in the filter window $w\{k\}$ will not be detected as noise because of its small $e(k)$ value.

In the present study, the feature vector is given by:

$$f\{k\} = \{c(k), l(k), e(k)\}. \tag{3}$$

The feature vector $f\{k\}$ serves as the input data set to the SVM impulse detector. This feature information is also used in the filtering stage.

3.2.2. *Training of the SVM classifier.* The optimal separating hyperplane can be obtained through a training process by using a set of supervised class labels $y_i \in \{-1, 1\}$ for the training corrupted image. The input in the training process is the set of unsupervised feature $f\{k\}$. Figure 2 shows the feedforward network architecture of the SVM impulse detector that identifies noise-free pixels and noise-corrupted pixels [1]. After learning, the nonlinearly inseparable discrimination function, as shown in Equation (7), is obtained to separate the training data into two classes (noise-free or noise-corrupted). That is, the optimal separating hyperplane is obtained. This efficient design minimizes the risk of misclassification not only in the training set (i.e., training errors) but also in the test set (i.e., generalization errors).

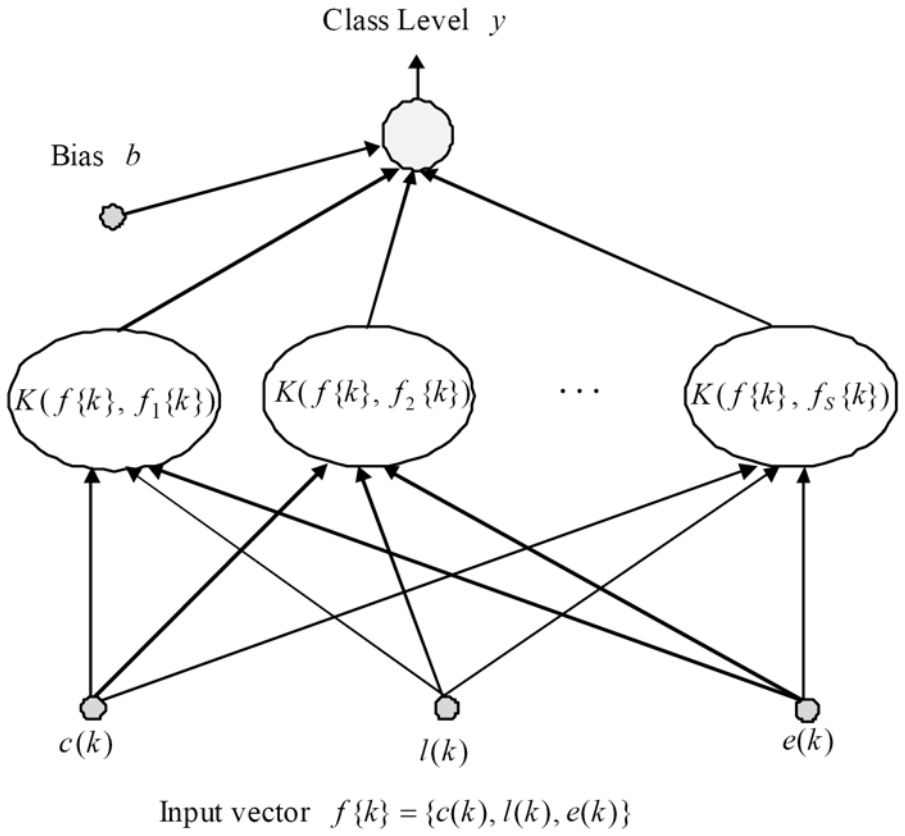


FIGURE 2. Feedforward network architecture of support vector machines

3.3. **Adaptive weight of the LUM filter.** Adaptive weight l allows the LUM filter to perform various degrees of noise suppression and image detail preservation. To determine adaptive weight l, a weight controller (the switching scheme shown in Figure 3) is proposed in this work [11]. Note that $d(\cdot)$ shown in Figure 3, which is defined as a function of the feature vector, is a classifier used to determine the partitioning, and $\beta_i(k), i \in \{1, 2, \ldots, M\}$ serves as adaptive weight l for the LUM filter. The scalar quantization (SQ) method is used to partition the feature vector space into M blocks, and the LMS learning algorithm is used to set the weight for each block to minimize the mean square error of the filter output.

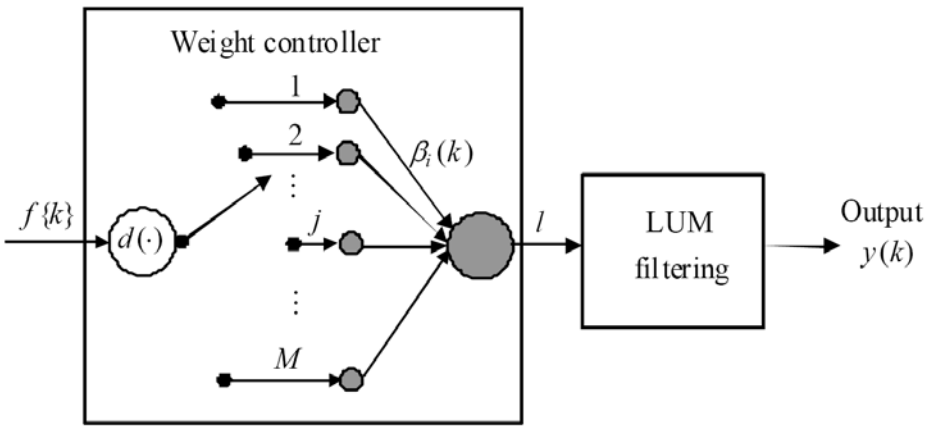


FIGURE 3. Structure of the adaptive LUM filter

3.3.1. *Partitioning of the feature space.* The feature vector exists in feature vector space $\Re^3$:

$$f\{k\} = \{c(k), l(k), e(k)\} \in \Re^3. \tag{4}$$

The weight controller, shown in Figure 3, decides that $\Re^3$ is partitioned into M mutually exclusive blocks Ω_i, $i = 1, 2, \ldots, M$. Then, each weight $\beta_i(k)$ is associated with the i-th block in the partition given by:

$$\Omega_i = \{f\{k\} \in \Re^3 : d(f\{k\}) = i\}, \quad i = 1, 2, \ldots, M, \tag{5}$$

where the classifier $d(\cdot)$ is now defined as a function of feature vector $f\{k\}$. As a result, the M blocks satisfy:

$$\Re^3 = \bigcup_{i=1}^{M} \Omega_i \text{ and } \Omega_i \cap \Omega_j = \phi, \text{ for } i \neq j. \tag{6}$$

Each input $x(k)$ corresponding to $f\{k\}$ is classified into one of the M blocks by the classifier $d(\cdot)$. Due to the low computational complexity of the partitioning indices, the classifier $d(\cdot)$ can be designed using simple scalar quantization (SQ) [9,11,31].

Each scalar component $f_j\{k\} \in \{c(k), l(k), e(k)\}$, $j = 1, 2, 3$ of $f\{k\}$ can be classified independently using SQ, which involves an encoder mapping process and a decoder mapping process [31]. The encoder mapping process involves receiving the input value $f_j\{k\}$ and providing an output codeword, which is determined using the interval in which the value falls. The decoder mapping process transforms the codeword into a representative value q. In the present study, the encoder mapping process divides the range $[0, 255]$ into five intervals such that each scalar component $f_j\{k\}$ belongs to one of the five intervals, as shown in Figure 4 [11]. This way, each block Ω_i can be represented by a Cartesian product of three interval blocks, s_1, s_2 and s_3; that is, $\Omega_i = s_1 \times s_2 \times s_3$. For example, an input $c(k)$ to the quantizer with a value between 2 and 5 will result in the output $q_1 = 2$. An input $l(k)$ with a value between 35 and 60 will result in the output $q_2 = 4$. An input $e(k)$ with a value between 60 and 255 will result in the output $q_3 = 5$. When the representation values q_1, q_2 and q_3 are obtained, a vector $V_i\{k\} = \{q_1, q_2, q_3\}$ is described as $d(f\{k\}) = i$, $i \in \{1, 2, \ldots, M\}$. Each unique vector $V_i\{k\}$ defines a distinct block Ω_i, and thus the feature vector space consists of M exclusive blocks that together form the entire observation vector space. Three-dimensional array A can be used to perform the partitioning as follows:

$$A[q_1][q_2][q_3], \quad 1 \leq q_1 \leq 5, \quad 1 \leq q_2 \leq 5, \quad 1 \leq q_3 \leq 5.$$

3.3.2. *Setting of weights using LMS algorithm.* Designing the optimal weight $\beta_i(k)$, $i = 1, 2, \ldots, M$ for the adaptive LUM filter requires minimizing the mean square error (MSE). The value of $\beta_i(k)$ can be obtained independently by executing the LMS algorithm, which is capable of minimizing the error function with respect to block Ω_i [26]. The weights $\beta_i(k)$ corresponding to block Ω_i can be adjusted in an iterative fashion along with the error surface toward the optimal solution. For each input $x(k)$ associated with block Ω_i, the value of $\beta_i(k)$ is updated iteratively in a gradient way:

$$\beta_i^{(t+1)}(k) = \begin{cases} \beta_i^{(t)}(k) - \eta_i^{(t)} |e(k)||x(k) - o(k)|, & \beta_i^{(t+1)} \geq 0 \\ 0, & \beta_i^{(t+1)} < 0 \end{cases}. \tag{7}$$

Here, the error $e(k)$ is the difference between the desired output $o(k)$ and the physical output $y(k)$ [11]. $\beta_i^{(0)}(k)$ represents the initial weight, and $\beta_i^{(t)}(k)$ is the weight after the t-th iteration. The learning rate $\eta_i^{(t)}$ at iteration t defines a decaying constant with the

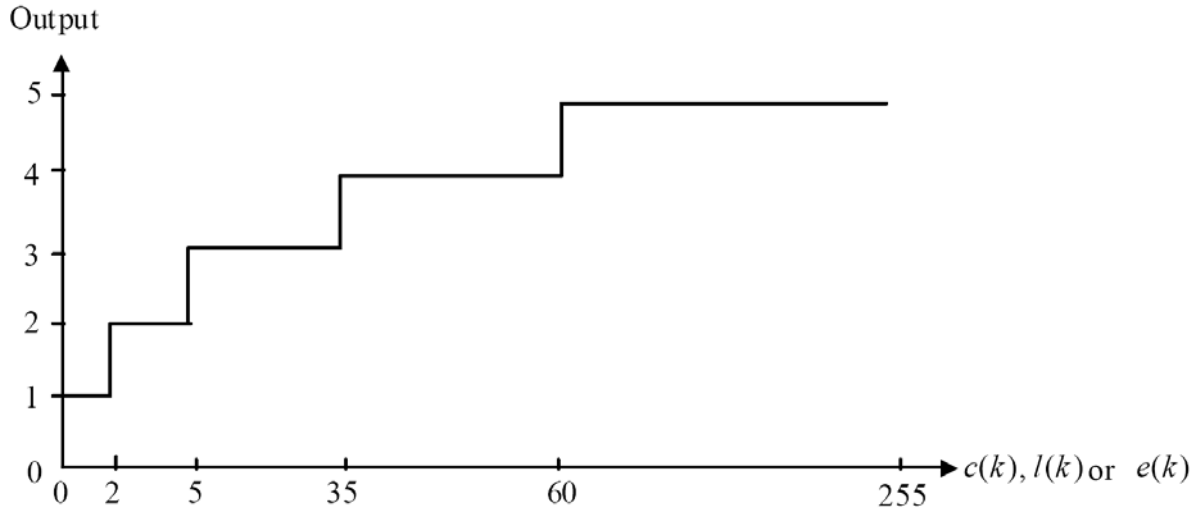


FIGURE 4. Quantizer input-output map for input scalar feature vector

iteration, such as $\eta_i^{(t)} = \alpha_0(1-\frac{t}{T})$, where α_0 is a predetermined constant and T is the total number of iterations involved. In order to improve the convergence speed, the algorithm iterates the process until the average difference of the distortion MSE falls below threshold θ, given by $\frac{|MSE^{(t)} - MSE^{(t-1)}|}{MSE^{(t)}} < \theta$, where t denotes the iteration number.

3.4. **Noise filtering.** The noise filtering of the ASVC filter is a combination of LUM filtering and adaptive LUM filtering, as shown in Figure 1.

3.4.1. *LUM filtering.* Feature vector $f\{k\}$ is computed from the test image, and then the SVM impulse detector classifies the input pixels as either noise-free or noise-corrupted according to the discrimination function learned in the training stage. If the input pixel is classified as impulsive noise by the SVM impulse detector, the LUM filter is activated and the pixel is replaced. Otherwise, the identity filter (no filtering) is activated to preserve the original intensity. Note that the inputs to the LUM filter here are selected from only the noise-free pixels decided by the SVM impulse detector within a filter window $w\{k\}$. Here, to reduce complexity, weight $l = (N+1)/2$ of the LUM filter is selected. That is, the LUM filter outputs the median value of the noise-free pixels in the filter window $w\{k\}$.

3.4.2. *Adaptive LUM filtering.* The SVM impulse detector might make mistakes. As a result, undetected noisy pixels may remain in the restored image and misdetected pixels may be mistakenly modified. Adaptive LUM filtering is thus incorporated in the ASVC filter to reduce the number of undetected and misdetected pixels. Since the ASVC filter adaptively selects an optimized weight to carry out the filtering operation for each input pixel $x(k)$ corresponding to the $\beta_i(k)$ of block Ω_i, better noise attenuation can be achieved without degrading the quality of fine details.

4. **Experimental Results.** Extensive experiments were conducted on a variety of 512×512 test images to evaluate the performance of the proposed ASVC filter. The peak

signal-to-noise ratio (PSNR) was employed to quantitatively measure the restoration performance, which is defined as:

$$PSNR = 10 \log_{10} \left(\frac{\sum_k 255^2}{\sum_k (o(k) - y(k))^2} \right) dB, \tag{8}$$

where 255 is the peak gray level of the image. In addition, the mean absolute error (MAE) was used as a quantitative measure to evaluate the levels of the edges and the details preserved, which is defined as:

$$MAE = \frac{\sum_k |o(k) - y(k)|}{H \times W}. \tag{9}$$

Larger PSNR values denote better image restoration, whereas smaller MAE values suggest better image detail preservation.

In this work, an impulsive noise model with noise ratio p is described as:

$$x(k) = \begin{cases} s(k), & \text{with probability } 1-p, \\ n(k), & \text{with probability } p, \end{cases} \tag{10}$$

where $s(k)$ and $n(k)$ represent original noise-free image pixel and the noise substitution for the original pixel, respectively [7]. There are two types of impulsive noise: fixed-valued and random-valued impulses. In 8-bit gray-scale images, fixed-valued impulsive noise, which is also known as salt and pepper noise, has equal probability of noise intensity at 0 and 255, whereas random-valued impulsive noise is uniformly distributed over the range of $[0, 255]$.

TABLE 1. Classification accuracy of SVM classifier

Image	Lena	F16	Boat	Lake	Goldhill	Cameraman
Fixed-valued impulse	0.9953	0.9870	0.9881	0.9812	0.9915	0.9829
Random-valued impulse	0.9767	0.9376	0.9728	0.9641	0.9698	0.9555

The optimal separating hyperplane was obtained using training image 'Couple' corrupted by 20% impulsive noise in the training process. The tested images were outside the training set to test the generalization capability. Table 1 shows the accuracy of the SVM impulse detector (or SVM classifier) for some images corrupted by 20% impulsive noise. Figure 5 shows the false noise detection using the SVM impulse detector.

3×3 filter windows were used in all the experiments. The optimal weight $\beta_i(k), i = 1, 2, \ldots, M$ was obtained using training image 'Couple' corrupted by 20% impulsive noise in the training process. The quantization interval values employed in the partitioning processes were obtained experimentally. $[0, 2)$, $[2, 5)$, $[5, 35)$, $[35, 60)$, $[60, 255]$ for variables $c(k), l(k)$ and $e(k)$ were found to be satisfactory interval values (shown in Figure 4) and thus used throughout the experiments.

Several experiments were conducted to compare the proposed ASVC filter with the standard median (MED) filter, the tri-state median (TSM) filter [6], switching scheme I (SWM-I) [16], the fuzzy median (FM) filter [4], the partition fuzzy median (PFM) filter [9], the fast peer group filter (FPGF) [15], and the adaptive two-pass median (ATM) filter [1] in terms of noise removal capability. Table 2 compares the PSNR and MAE results of removing both fixed-valued and random-valued impulsive noise at 20%. As the table shows, the proposed ASVC filter significantly outperforms the other schemes. Figure 6 shows the restoration results comparison for the image 'Cameraman' corrupted by 20% random-valued impulsive noise among MED, FM, ATM and ASVC. The ASVC filter

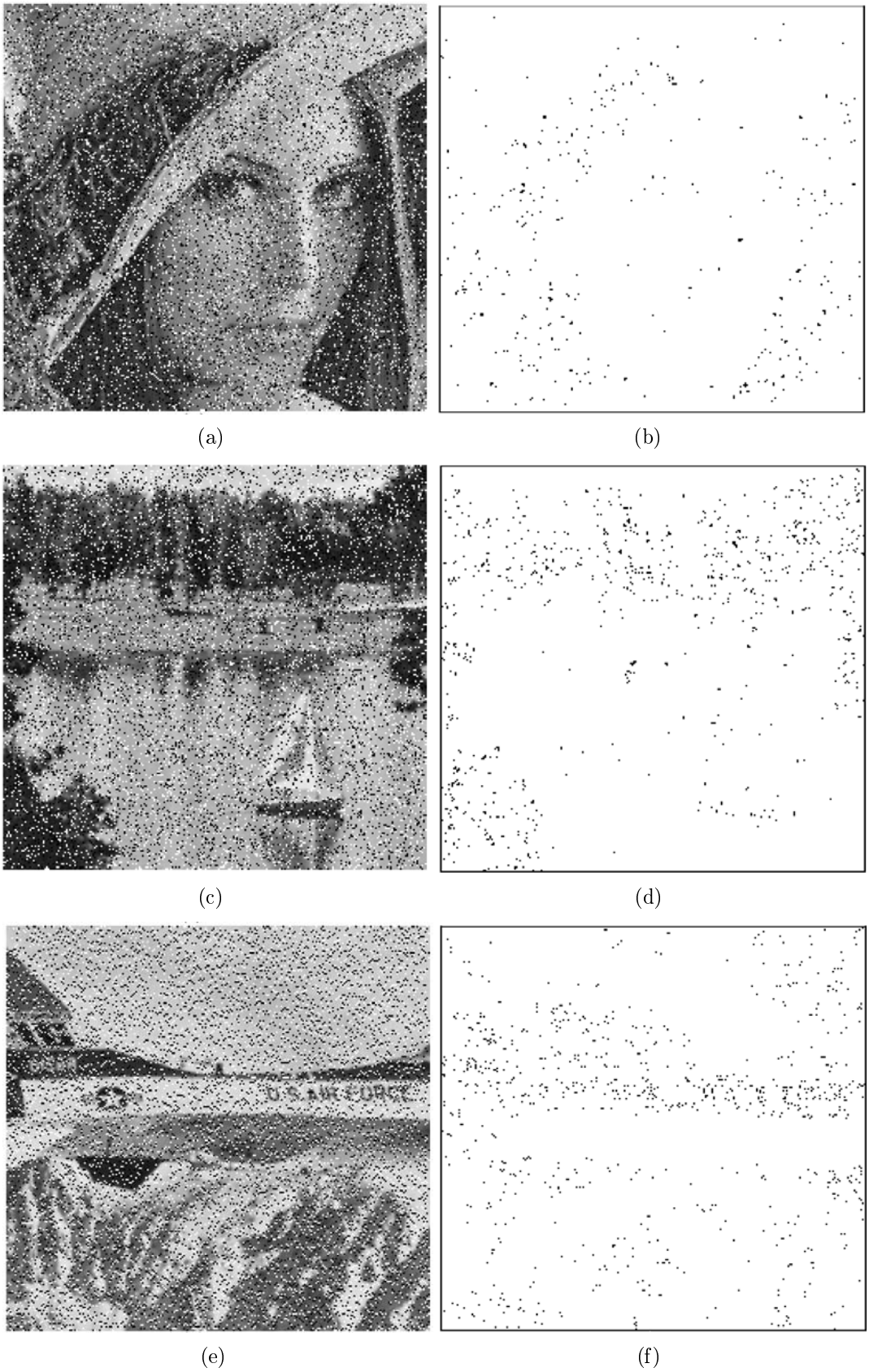
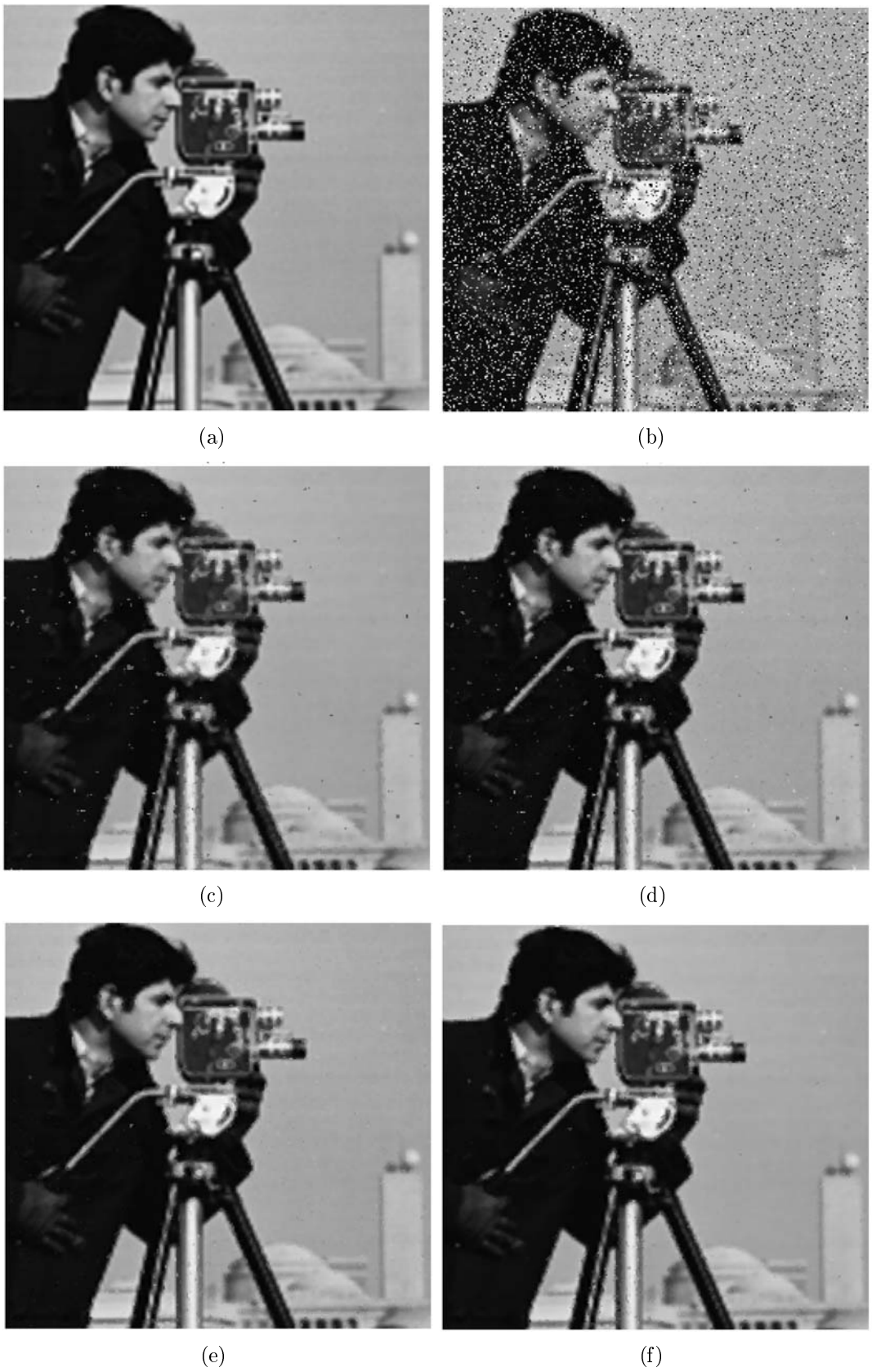


FIGURE 5. Noise identification. (a) 'Lena', (c) 'Lake' and (e) 'F16', each corrupted by 20% fixed- valued impulsive noise, (b) False detection for 'Lena', (d) False detection for 'Lake', (f) False detection for 'F16'.



(a) (b)

(c) (d)

(e) (f)

FIGURE 6. Subjective visual quality of restored image 'Cameraman'. (a) Original image, (b) image corrupted by 20% random-valued impulsive noise, and images filtered by (c) MED filter, (d) FM filter, (e) ATM filter and (f) ASVC filter.

TABLE 2. Restoration result comparison of PSNR (dB) and MAE for 20% (a) fixed-valued and (b) random-valued impulsive noise

(a)

Filter	Image											
	Lena		F16		Boat		Lake		Goldhill		Cameraman	
	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE
MED	30.18	3.44	29.62	3.47	29.20	3.89	27.19	5.61	28.84	4.83	33.76	1.85
TSM	31.84	2.75	32.76	2.82	31.16	2.97	29.73	3.51	31.55	2.85	34.71	1.42
SWM-I	31.64	1.58	32.43	1.18	30.26	2.06	28.55	2.69	30.58	2.15	34.64	1.19
FM	31.32	1.88	31.45	1.94	30.86	2.16	28.61	3.14	30.95	2.39	34.82	1.14
PFM	35.52	1.57	32.92	1.54	33.34	1.82	31.13	2.52	33.87	2.01	36.08	0.99
FPGF	31.06	1.82	29.66	2.16	29.94	2.29	27.58	3.38	30.30	2.36	34.11	1.16
ATM	35.77	1.46	32.99	1.45	33.05	1.76	30.62	2.47	33.06	1.98	36.64	0.83
ASVC	36.46	1.36	33.94	1.35	34.27	1.64	31.30	2.28	34.08	1.84	37.17	0.98

(b)

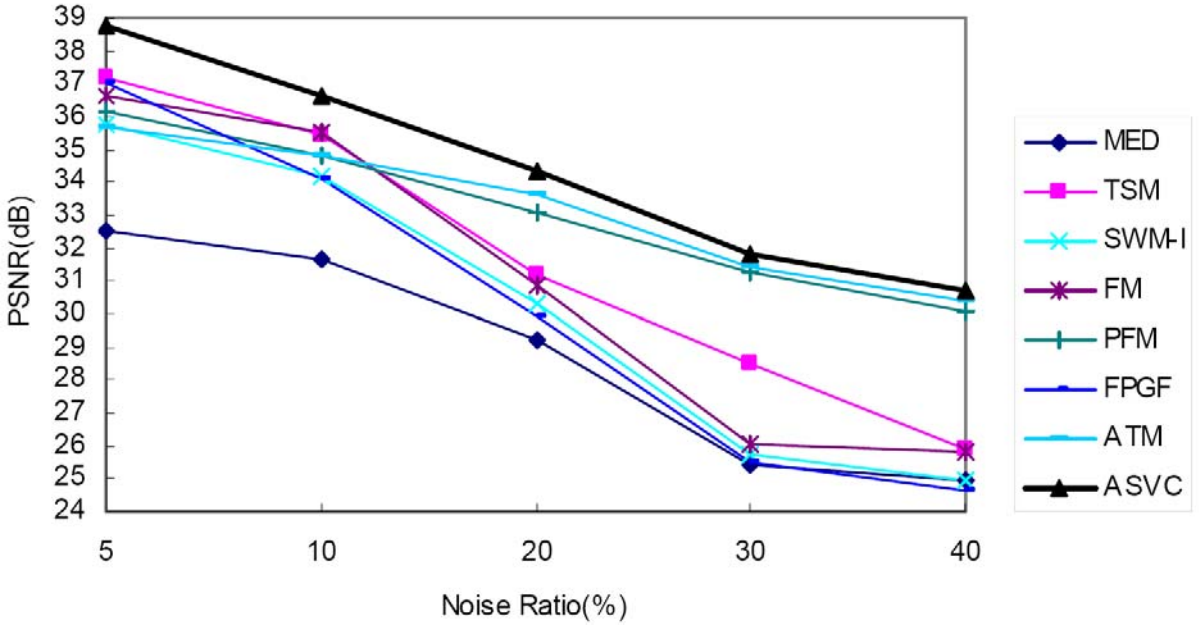
Filter	Image											
	Lena		F16		Boat		Lake		Goldhill		Cameraman	
	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE	PSNR	MAE
MED	31.72	3.46	30.93	2.95	30.14	3.92	27.84	5.71	29.71	4.84	31.51	2.31
TSM	34.03	1.91	31.56	2.77	32.29	2.83	30.22	3.26	32.44	2.61	31.62	1.89
SWM-I	32.08	2.09	29.62	2.16	30.78	2.35	28.86	3.04	31.01	2.46	31.14	1.63
FM	33.40	2.11	30.79	2.11	32.11	2.37	29.76	3.23	32.22	2.63	32.15	1.61
PFM	33.88	2.03	32.03	1.95	32.43	2.32	30.11	3.26	32.44	2.61	34.04	1.36
FPGF	30.11	2.69	28.52	2.76	29.46	2.87	28.15	3.46	29.67	2.92	29.65	2.22
ATM	34.26	1.64	32.00	1.69	32.42	1.94	29.96	2.74	32.26	2.19	33.34	1.46
ASVC	34.63	1.59	32.20	1.54	32.86	1.84	30.43	2.49	32.73	2.09	34.26	1.14

produces a restored image with the best subjective visual quality by offering more noise suppression and detail preservation.

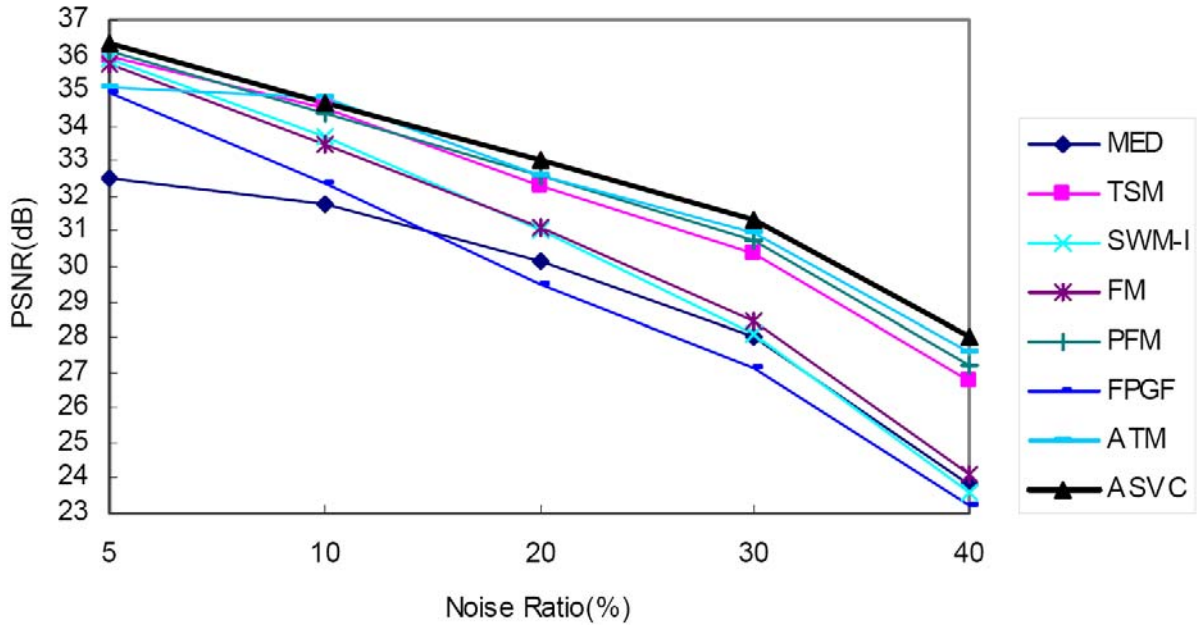
The robustness of the ASVC filter was also tested. The trained discrimination function and the optimal weights of block Ω_i were independent of the noise intensity in all experiments. Figure 7 shows the PSNR comparison with restored image 'Boat' initially corrupted by 5% to 40% fixed-valued or random-valued impulsive noise. As shown in the figure, the ASVC filter outperforms the other techniques in terms of robustness, even though the 20% impulsive noise training image 'Couple' is independent of the actual corruption percentage in filtering.

5. **Conclusion.** A novel decision-based median filter based on SVMs was developed to preserve image details while effectively suppressing impulsive noise. An impulse detector design based on SVMs is responsible for judging whether the input pixel is noisy. In addition, an adaptive LUM filter was proposed to efficiently improve the detection error rate of the SVM impulse detector. The excellent generalization capability of SVMs and the optimal weight of each block allow the mean square error of the filter output to be minimized. The experimental results demonstrate that the proposed ASVC filter is superior to a number of well-accepted decision-based median filters. The ASVC filter is also capable of giving satisfactory perceptual quality.

Acknowledgements. The authors would like to thank Prof. Chih-Jen Lin for providing LIBSVM software, a library for support vector machines (version 2.83) and acknowledge the National Science Council of Taiwan for financially supporting this research under grant NSC 99-2628-E-274-011.



(a) Fixed-valued impulsive noise



(b) Random-valued impulsive noise

FIGURE 7. Restoration result comparison of PSNR (dB) for filtering 'Boat' image corrupted by impulsive noise

REFERENCES

[1] T.-C. Lin and P.-T. Yu, Adaptive two-pass median filter based on support vector machines for image restoration, *Neural Computation*, vol.16, pp.333-354, 2004.

[2] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, CRC, Boca Raton, FL, 1997.

[3] R. Lukac, B. Smolka, K. Martin, K. N. Plataniotis and A. N. Venetsanopoulos, Vector filtering for color imaging, *IEEE Signal Processing Magazine*, vol.22, pp.74-86, 2005.

[4] K. Arakawa, Median filters based on fuzzy rules and its application to image restoration, *Fuzzy Sets and Systems*, vol.77, pp.3-13, 1996.
[5] J.-G. Camarena, V. Gregori, S. Morillas and A. Sapena, Some improvements for image filtering using peer group techniques, *Image and Vision Computing*, vol.28, pp.188-201, 2010.
[6] T. Chen, K. K. Ma and L. H. Chen, Tri-state median filter for image denoising, *IEEE Transactions on Image Processing*, vol.8, pp.1834-1838, 1999.
[7] T. Chen and H. R. Wu, Application of partition-based median type filters for suppressing noise in images, *IEEE Transactions on Image Processing*, vol.10, pp.829-836, 2001.
[8] A. Hussain, M. A. Jaffar, A. M. Mirza and A. Chaudhry, Detail preserving fuzzy filter for impulse noise removal, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(B), pp.3583-3592, 2009.
[9] T.-C. Lin and P.-T. Yu, Partition fuzzy median filter based on fuzzy rules for image restoration, *Fuzzy Sets and Systems*, vol.147, pp.75-97, 2004.
[10] T.-C. Lin and P.-T. Yu, Impulse noise detection and removal using multiple thresholds for image restoration, *Journal of Information Science and Engineering*, vol.22, pp.189-198, 2006.
[11] T.-C. Lin, A new adaptive center weighted median filter for suppressing noise in images, *Information Sciences*, vol.177, pp.1073-1087, 2007.
[12] T.-C. Lin, Partition belief median filter based on Dempster-Shafer theory in image processing, *Pattern Recognition*, vol.41, pp.139-151, 2008.
[13] P. K. Sa and B. Majhi, An improved adaptive impulsive noise suppression scheme for digital images, *International Journal of Electronics and Communications*, vol.64, pp.322-328, 2010.
[14] S. Schulte, V. D. Witte, M. Nachtegael, D. V. Weken and E. E. der Kerre, Fuzzy random impulse noise reduction method, *Fuzzy Sets and Systems*, vol.158, pp.270-283, 2007.
[15] B. Smolka and A. Chydzinski, Fast detection and impulsive noise removal in color images, *Real-Time Imaging*, vol.11, pp.389-402, 2005.
[16] T. Sun and Y. Neuvo, Detail-preserving median based filters in image processing, *Pattern Recognition Letter*, vol.15, pp.341-347, 1994.
[17] S. J. Ko and Y. H. Lee, Center weighted median filters and their applications to image enhancement, *IEEE Transactions on Circuits and Systems*, vol.38, pp.984-993, 1991.
[18] L. Yin, R. Yang, M. Gabbouj and Y. Neuvo, Weighted median filters: A tutorial, *IEEE Transactions on Circuits and Systems*, vol.43, pp.157-192, 1996.
[19] C. Kotropoulos and I. Pitas, Adaptive LMS L-filters for noise suppression in images, *IEEE Transactions on Image Processing*, vol.5, pp.1596-1609, 1996.
[20] S. Ohno and H. Sakai, Convergence behavior of the LMS algorithm in subband adaptive filtering, *Signal Processing*, vol.81, pp.1053-1059, 2001.
[21] R. Lukac and S. Marchevsky, Boolean expression of LUM smoothers, *IEEE Signal Processing Letters*, vol.8, pp.292-294, 2001.
[22] R. Lukac, Binary LUM smoothing, *IEEE Signal Processing Letters*, vol.9, pp.400-403, 2002.
[23] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, USA, 1995.
[24] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, USA, 1998.
[25] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.
[26] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice- Hall, 1999.
[27] C.-C. Chang and C.-J. Lin, Training nu-support vector regression: Theory and algorithms, *Neural Computation*, vol.14, pp.1959-1977, 2002.
[28] C.-W. Hsu and C.-J. Lin, A simple decomposition method for support vector machines, *Machine Learning*, vol.46, pp.291-314, 2002.
[29] C.-J. Lin, A formal analysis of stopping criteria of decomposition methods for support vector machines, *IEEE Transactions on Neural Networks*, vol.13, pp.1045-1052, 2002.
[30] K. M. Lin and C. J. Lin, A study on reduced support vector machines, *IEEE Transactions on Neural Networks*, vol.14, pp.1449-1459, 2003.
[31] K. Sayood, *Introduction to Data Compression*, 2nd Edition, Morgan Kaufman, 2000.
[32] M.-H. Tsai, Y.-B. Lin and C.-M. Wang, Image sharing with steganography and cheater identification, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(A), pp.1165-1178, 2010.
[33] C.-H. Lin, Y.-C. Li, H.-F. Chien and S.-L. Chien, Multipurpose image authentication method based on vector quantization, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(B), pp.1435-1446, 2010.

[34] G. Ye, A novel logistic-based image encryption scheme, *ICIC Express Letters*, vol.4, no.3(B), pp.979-984, 2010.
[35] X. Guo, H. Zhang and Z. Chang, Image thresholding algorithm based on image gradient and fuzzy set distance, *ICIC Express Letters*, vol.4, no.3(B), pp.1059-1064, 2010.
[36] T.-C. Lin and C.-M. Lin, Decision-based adaptive low-upper-middle filter for image processing, *International Journal of Innovative Computing, Information and Control*, vol.7, no.10, pp.5977-5990, 2011.