

## Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

Bae-Muu Chang<sup>1,2</sup>, Hung-Hsu Tsai<sup>\*3</sup>, Xuan-Ping Lin<sup>3</sup>, and Pao-Ta Yu<sup>1</sup>

<sup>1</sup> Department of Computer Science and Information Engineering,  
National Chung Cheng University, Min-Hsiung, Chia-Yi 621, Taiwan, R.O.C.  
bmchang@cs.ccu.edu.tw (B.-M. Chang)  
cspity@cs.ccu.edu.tw (P.-T. Yu)

<sup>2</sup> Department of Information Management,  
Chienkuo Technology University, Chang-Hua, Chang-Hua 500, Taiwan, R.O.C.  
bmchang@cc.ctu.edu.tw

<sup>3</sup> Department of Information Management,  
National Formosa University, Hu-Wei, Yun-Lin 632, Taiwan, R.O.C.  
thh@nfu.edu.tw (H.-H. Tsai)  
no300601@gmail.com (X.-P. Lin)

**Abstract.** This paper proposes the median-type filters with an impulse noise detector using the decision tree and the particle swarm optimization, for the recovery of the corrupted gray-level images by impulse noises. It first utilizes an impulse noise detector to determine whether a pixel is corrupted or not. If yes, the filtering component in this method is triggered to filter it. Otherwise, the pixel is kept unchanged. In this work, the impulse noise detector is an adaptive hybrid detector which is constructed by integrating 10 impulse noise detectors based on the decision tree and the particle swarm optimization. Subsequently, the restoring process in this method respectively utilizes the median filter, the rank ordered mean filter, and the progressive noise-free ordered median filter to restore the corrupted pixel. Experimental results demonstrate that this method achieves high performance for detecting and restoring impulse noises, and outperforms the existing well-known methods.

**Keywords:** Impulse noise detector, Decision tree, Particle swarm optimization, Median-type image filter, Noise removal.

### 1. Introduction

Digital images are sometimes contaminated by impulse noises during transmission processes. These noises usually degrade the image processing results severely such as edge detection, image segmentation, and object recognition. Thus, it is urgently essential to remove noises before the image

processing procedures are performed. For this purpose, several well-known filters have been presented in the past years. Nonlinear filters are usually better than linear filters because of their good impulse noise removal and edge preservation [1]. The median filter which is a nonlinear filter is the popular well-known and the most often employed for removing impulse noises [1]. Although it is effective in filtering impulse noises, the median filter still blurs some fine details and often damages edges when it is applied to the digital images uniformly.

Hence, various modified median-based filters have been proposed to enhance the typical median filter in the recent years, such as the weighted median (WM) filter [2], the tri-state median (TSM) filter [3], the progressive switching median (PSM) filter [4], the center weighted median (CWM) filter [13], the adaptive center weighted median (ACWM) filter [5], the multistate median (MSM) filter [6], the fuzzy-rule-based median (FM) filter, and the iterative median (IM) filter [7]. Although these filters achieve better detection and restoration results, they still tend to blur some fine details. In order not to damage good pixels, the decision-based median filters with switching schemes have been proposed in some papers [3, 13].

In addition, the directional weighted median (DWM) filter [8] performs well at noise ratio higher than 30% when it is compared with the TSM, the ACWM, and the MSM filters. Nevertheless, it needs to calculate some fixed parameters to perform image enhancement. Moreover, the high performance detection (HPD) filter employs the sufficient similar neighbor criteria for image restoration [9]. However, it also needs some fixed parameters, such as the intensity value and the threshold, to work correctly. Thus, the HPD filter cannot completely separate the noise-free pixels from the noise-corrupted pixels, either.

In the recent years, most proposed algorithms for selective impulse noise detection need the thresholds to classify the input pixel as either noise-corrupted or noise-free. These thresholds will severely affect the performance for noise detection. They are finally determined by a series of the iterative experiments.

The decision tree (DT) is popularly used in the field of data mining. Especially, it is a very efficient method in classification problems. In image enhancement, the purpose for noise detection is detecting the pixels whether they are corrupted or not. Thus, noise detection will be strongly regarded as a classification problem. In the past years, some well-known methods need the thresholds for classification which are obtained by manual or forced-searching approaches. They are time-consuming, inefficient, and high time complexity. Therefore, it is essential to employ a systematic approach for solving this problem. In this paper, the particle swarm optimization (PSO) is utilized to optimize the thresholds to find out the approximate optimized DT and the suboptimal solutions for a set of these parameters. Hence, employing the DT is more accurate for noise detection.

The PSO derives next generation using the error values in each generation. It is easier to find the nearer optimal solution using the PSO. The main reason is that the PSO calculates the vector of movement and derives

## Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

the position for next generation by considering the optimal solution from the first generation to the current generation. Therefore, the PSO has high ability with memory [17].

In this paper, the novel selective image filters called the median-type filters with an impulse noise detector (IND) using the DT and the PSO for image restoration (MDP), is proposed. It integrates the features of 10 impulse noise detection algorithms based on the DT to construct an adaptive hybrid IND. Also, it employs the PSO to well determine the optimized thresholds for the hybrid noise detector.

The remainder of this paper is arranged as follows. Basic concept is stated in Section 2. Principles for the DT and the PSO are then in detail described in Section 3. In Section 4, the design for the MDP filter will be next depicted. The training structure diagram for the proposed impulse noise detector in the MDP filter is shown in this section. Subsequently, the experimental results demonstrate the comparison for the PSNR values and the restored images at various noise ratios in some well-known corrupted images for different methods in Section 5. Finally, conclusion is given in Section 6.

## 2. Basic Concept

A gray-level image is represented as a two-dimensional  $L \times K$  matrix  $X = \{x_{ij} \mid 1 \leq i \leq L, 1 \leq j \leq K\}$ , where  $L$  and  $K$  are its height and width, respectively, and  $x_{ij} \in \{0, 1, 2, \dots, 255\}$  is the pixel gray-level value at position  $(i, j)$  in  $X$ .

A filter window with size  $S = (2\tau + 1)^2 = 2n + 1$  will slide over the image  $X$  at position  $(i, j)$  to formulate a sample matrix  $X_{ij}$ , where  $1 \leq i \leq L$ ,  $1 \leq j \leq K$ , and  $S$  is generally an odd number. Let the value for the central pixel in the filter window  $X_{ij}$  be  $x_{ij}$ . The filter window  $X_{ij}$  usually slides over the image  $X$  from left to right and top to bottom. For better clarity, the sample matrix  $X_{ij}$  can be rewritten as  $X_{ij} = (x_{i-\tau, j-\tau}, x_{i-\tau, j-\tau+1}, \dots, x_{ij}, \dots, x_{i+\tau, j+\tau-1}, x_{i+\tau, j+\tau})$ .

In order to change the  $(2\tau + 1) \times (2\tau + 1)$  filter window into a one-dimensional vector, it is reorganized by  $\mathbf{x}(k) = (x_{-n}(k), \dots, x_{-1}(k), x_0(k), x_1(k), \dots, x_n(k))$ , where  $x_0(k)$  (or  $\mathbf{x}(k)$ ) is the original central pixel value at location  $k$ . For instance, a  $3 \times 3$  filter window centered at  $x_0(k)$  is considered, such that  $\mathbf{x}(k) = (x_{-4}(k), \dots, x_{-1}(k), x_0(k), x_1(k), \dots, x_4(k))$ , where  $k = (i - 1) \times K + j$  indicates the pixel located at position  $(i, j)$  in the image  $X$ , and  $x_0(k)$  (or  $\mathbf{x}(k)$ ) stands for the central pixel in the filter window.

The general output value in the MF filter with the filter window sized  $2n + 1$  is

$$y(k) = \text{median}(\mathbf{x}(k))$$

described as  $= \text{median}(x_{-n}(k), \dots, x_{-1}(k), x_0(k), x_1(k), \dots, x_n(k))$ , where *median* represents the median operation.

### 3. Principles for the DT and the PSO

#### 3.1. Decision Tree

The DT is a kind of tree structure which is applied in classification problems [16]. It can automatically classify the data according to the splitting condition. The DT can deal with variables with continuous type or category type. Its model can sufficiently show each variable's relative importance and effectively deal with a huge dataset with many variables [16].

The purpose for noise detection is detecting the pixels whether they are corrupted or not when the images are processed. Noise detection is certainly considered as a classification problem. Thus, it is necessary to employ a systematic approach for solving this problem. In this paper, the algorithm for classification and regression trees (CART) is employed to construct a DT for the IND in noise detection. Experimentally, employing the DT is more accurate in noise detection.

#### 3.2. Particle Swarm Optimization

Kennedy proposed the PSO algorithm in 1995 [17]. It is designed according to birds' path of movement for food searching. The algorithm integrates the relationship between birds' individual and group features. Birds will decide the next direction and the distance of movement by referencing the past directions of movement and the current position when they search for foods. A particle in the PSO algorithm is simulated as an individual for a bird's food searching. The feature that a particle is simulated as a biological individual in the hyper-dimensional search space is employed to search for the optimal solution. Each particle will simulate the psychological tendencies from every other individual in the group of society. Apart from the individual direction searching, each individual will also learn from the best individual in the group of society. By the way of mutual learning, the individual will be able to find out the place for more food. In other words, the particles can therefore search for better solutions [17].

Here, the detailed procedure for the PSO is described as follows:

1. Set the initial parameters

The parameters swarm size, weight, range of movement for particles, and the number for training generations, must be initially set. The swarm size indicates the number of the particles in each generation. The weight represents one of the parameters employed in calculating the movement vector. The number for training generations means the number for training.

2. Get the fitness

The method for defining the fitness function can be designed by a function of the error value or the accuracy rate. The returned fitness values, such as the

## Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

error value and the accuracy rate, are usually the ultimate goal in a problem. Therefore, the fitness can be utilized to decide the merit or demerit of the current location for the particles, and can be employed to determine whether to continue training.

### 3. The conditions to stop training

Here, two conditions for stop of training will be listed. First, the maximum number of generations for training is reached. Second, the fitness value has satisfied the problem's requirements.

### 4. Get Gbest and Pbest

The current fitness value for each particle is compared with that of each individual best position. If the particle's fitness value is less than that of its Pbest, the individual best position will be replaced with the particle current position. Similarly, its fitness value is compared with that of its Gbest. If the particle's fitness value is less than that of its Gbest, the position of Gbest will be substituted with that of the particle. Here, getting Gbest and Pbest is described in Eq. (1).

$$\begin{aligned} \text{if } \text{fitness}(\mathbf{X}_i) < \text{fitness}(\text{Pbest}_i) \text{ then } \text{Pbest} = \mathbf{X}_i(t), \\ \text{if } \text{fitness}(\mathbf{X}_i) < \text{fitness}(\text{Gbest}) \text{ then } \text{Gbest} = \mathbf{X}_i(t), \end{aligned} \quad (1)$$

where  $t$  represents the  $t$ th generation,  $i=1, \dots, n$ ,  $n$  indicates the number of the particles,  $\mathbf{X}_i = (X_{i1}, X_{i2}, \dots, X_{im}) \in \mathbb{R}^m$  stands for the particle current position,  $m$  denotes the particle dimensionality,  $\text{Pbest}_i \in \mathbb{R}^m$  represents the best position for the  $i$ th individual particle's path of movement,  $\text{Gbest} \in \mathbb{R}^m$  indicates the position closest to the optimal solution in the group, and fitness means the fitness function.

### 5. Calculate the movement vector

The movement vector is defined as follows:

$$\mathbf{V}_i(t+1) = w\mathbf{V}_i(t) + c_1 \times r_1 \times (\text{Pbest}_i - \mathbf{X}_i(t)) + c_2 \times r_2 \times (\text{Gbest} - \mathbf{X}_i(t)), \quad (2)$$

where  $\mathbf{V}_i = (V_{i1}, V_{i2}, \dots, V_{im}) \in \mathbb{R}^m$  is the movement vector for the  $i$ th particle,  $w$  indicates the inertia weight,  $c_1$  and  $c_2$  are the acceleration coefficients which are random numbers in the interval  $[0, 1]$ ,  $r_1$  and  $r_2$  are also random numbers in the interval  $[0, 1]$ , the first part  $w\mathbf{V}_i(t)$  represents the particle's inertia, the second part  $c_1 \times r_1 \times (\text{Pbest}_i - \mathbf{X}_i(t))$  stands for the particle's cognition-only model, and the third part  $c_2 \times r_2 \times (\text{Gbest} - \mathbf{X}_i(t))$  denotes the particle's social-only model.

### 6. Modify the particle's position

The method for modifying the particle's position is defined as follows:

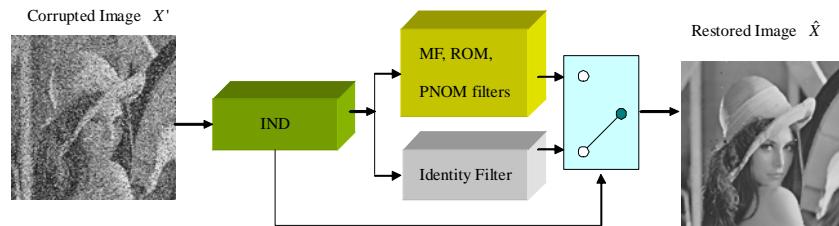
$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) + \mathbf{V}_i(t+1). \quad (3)$$

The particle  $i$ 's  $(t+1)$ th generation position is the vector addition of the  $t$ th generation position and the  $(t+1)$ th generation's movement vector. That is, adding the particle's current position and its movement vector becomes the particle's new position.

## 4. Design for the MDP Filter

### 4.1. The Structure for the MDP Filter

The IND employs the DT and the PSO algorithm to integrate 10 noise detection algorithms to find out the positions for the corrupted pixels in the images. Fig. 1 demonstrates the structure for the MDP image filter. In this paper, the MDP filter replaces the pixel with the modified median filters' results employing the MF, the ROM, and the PNOM filters, when the pixel is considered as noise-corrupted. Otherwise, the pixel will be kept unchanged.



**Fig. 1.** The structure for the MDP image filter

### 4.2. The Employed Noise Detection Algorithms

Generally, the values for most pixels which are corrupted by impulse noise are more eminent than those for fine pixels. Therefore, the difference between the input pixel  $x(k)$  and the median value  $\text{median}(x(k))$  in the filter window shows a sufficient reason to determine a corrupted pixel [10]. Here, 10 feature variables are employed as 10 impulse noise detectors to respectively estimate the feature value for each pixel. For clarity,  $x(k)$  and  $x_0(k)$  will be used interchangeably throughout this paper. Also, the filter window  $X_{ij}$  and one-dimensional vector  $x(k)$  will be employed alternatively.

**Detector 1:** the difference between the central pixel value and the median pixel value

The variable  $u(k)$  specifies the absolute difference between the input pixel  $x(k)$  and the median value  $\text{median}(x(k))$  as follows:

$$u(k) = |x(k) - \text{median}(x(k))|, \quad (4)$$

## Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

where median indicates the median operation. The variable  $u(k)$  proposes a simple method for detecting impulse noise. A large  $u(k)$  value which is greater than the threshold  $T_u$  represents that the input  $x(k)$  is dissimilar to the median value  $\text{median}(x(k))$  in the filter window  $x(k)$ ; that is, it strongly suggests the central pixel  $x(k)$  is corrupted by impulse noise.

**Detector 2:** the difference between the individual pixel value and the average of the sample pixels' values

The variable  $p_j(k)$  of the local contrast at location  $k$  in the filter window  $x(k)$  is defined as follows:

$$p(k) = \frac{|x(k) - \bar{x}(k)|}{\sum_{i=-n}^n |x_i(k) - \bar{x}(k)|}, \quad (5)$$

where  $\bar{x}(k)$  represents the average gray-level value in the filter window  $x(k)$  with size  $2n+1$ . A large  $p(k)$  value which is greater than the threshold  $T_p$  indicates that it heavily suggests the central pixel  $x(k)$  is corrupted by impulse noise.

Nevertheless, if only variables  $u(k)$  and  $p(k)$  are employed to determine whether the input  $x(k)$  is corrupted or not, then it is difficult to completely detect impulse noise [17]. For instance, a line component usually exists in the image and its width is only one pixel; hence, if the input pixel  $x(k)$  is located on the line, it may be identified as an impulse noise and be removed [10]. In order to prevent bad judgments, it is essential to utilize other observations to enhance the detecting correctness, thus  $v(k)$  and  $q(k)$  are proposed as follows:

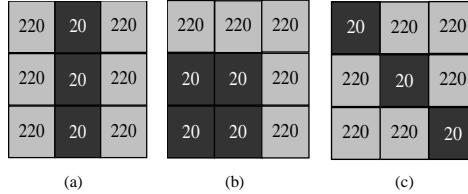
**Detector 3:** the average of the differences between the central pixel value and two closest pixels' values

If only variable  $u(k)$  is employed to determine whether the input pixel  $x(k)$  is corrupted or not, then it will mistakenly detect  $x(k)$  as noise-corrupted in Fig. 2(a). In order to remove the weakness, the variable  $v(k)$  is further employed for better noise detection in Eq. (6).

$$v(k) = \frac{|x(k) - x_{\beta_1}(k)| + |x(k) - x_{\beta_2}(k)|}{2}, \quad (6)$$

where  $|x(k) - x_{\beta_1}(k)| \leq |x(k) - x_{\beta_2}(k)| \leq |x(k) - x_i(k)|$ ,  $-n \leq i \leq n, i \neq \beta_1, \beta_2$ . The pixel values for  $x_{\beta_1}(k)$  and  $x_{\beta_2}(k)$  are closest to that of  $x(k)$  in the filter window  $x(k)$ . A large  $v(k)$  value which is greater than the threshold  $T_v$  indicates that it strongly suggests the central pixel  $x(k)$  is corrupted by impulse noise. If we employ  $v(k)$  as a detector, then the pixels on the line component in the filter

window  $x(k)$  will not be detected as impulse noises because of the small  $v(k)$  value [10, 15, 17].



**Fig. 2.** The filter window  $x(k)$  with size  $3 \times 3$  (a) the line component (b) the edge component and (c) the thin line component in the filter window, respectively

**Detector 4:** the difference between the central pixel value and the median value of the filter window  $x(k)$  with the repeating central pixels

In Fig. 2(b), the input pixel  $x(k)$  is mistakenly considered as noise-corrupted by  $u(k)$  when it is on the edge component in the filter window  $x(k)$ . To get good judgment, the variable  $q(k)$  is utilized for enhancement. It is described as follows:

$$q(k) = |x_0(k) - c^s(k)|, \quad (7)$$

$$\text{where } c^s(k) = \text{median} \left( x_{-n}(k), \dots, x_{-1}(k), \underbrace{x_0(k), x_0(k), x_0(k)}_{s \text{ times}}, \dots, x_n(k) \right), \quad (8)$$

and  $s$  indicates a positive integer for the repeating times [5, 11].

A large  $q(k)$  value which is greater than the threshold  $T_q$  indicates that it heavily suggests the central pixel  $x(k)$  is corrupted by impulse noise. In [10], Lin presented that any pixel on the edge and the line components in the filter window  $x(k)$  will not be detected as noise-corrupted if the variables  $q(k)$  and  $v(k)$  are employed. Thus, the system can correctly determine that no impulse noise is located at the pixel  $x(k)$  because of the small  $q(k)$  and  $v(k)$  values. A small  $q(k)$  value represents that it heavily suggests the central pixel  $x(k)$  is not corrupted by impulse noise, in Fig. 2(b).

**Detector 5:** the difference between the central pixel value and the average of the sample pixels' values

The variable  $g(k)$  is defined as follows:

$$g(k) = |x(k) - \text{avg}(x(k))|, \quad (9)$$

where  $\text{avg}(x(k)) = \frac{1}{2n} \sum_{i=-n, i \neq 0}^n x_i(k)$ , it denotes the average of all pixels' values in the filter window  $x(k)$  without  $x_0(k)$ . A large  $g(k)$  value which is greater than the threshold  $T_g$  indicates that a big difference between the input pixel

## Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

$x(k)$  value and the average gray-level value in the filter window  $\mathbf{x}(k)$ , and the central pixel  $x(k)$  is possibly corrupted by impulse noise [10].

**Detector 6:** the average of differences between the central pixel value and four closest pixels' values

The variable  $d(k)$  is defined as follows:

$$d(k) = \frac{\sum_{i=1}^4 |x_0(k) - x_{\beta_i}(k)|}{4}, \quad (10)$$

where  $|x_0(k) - x_{\beta_j}(k)| \leq |x_0(k) - x_{\beta_{j+1}}(k)|$ ,  $j = 1, 2, 3$ ,  $x_{\beta_1}, x_{\beta_2}, x_{\beta_3}$ , and  $x_{\beta_4}$  are the first four pixels in the filter window  $\mathbf{x}(k)$  which are nearest to  $x(k)$ . A large  $d(k)$  value which is greater than the threshold  $T_d$  indicates that it heavily suggests the central pixel  $x(k)$  is corrupted by impulse noise because the difference between  $x(k)$  value and its four nearest pixels' values is too big, respectively [10]. Here, although  $d(k)$  can effectively detect impulse noise in the image, the thin line components in the filter window  $\mathbf{x}(k)$  will be mistakenly detected as noise-corrupted in Fig. 2(c).

**Detector 7:** the noise detection employing the edge detection median (EDM) filter

The variable  $u(k)$  will mistakenly detect the input pixel  $x(k)$  as noise-corrupted which is on the thin line component in the filter window  $\mathbf{x}(k)$ . Thus, Zhang proposed the EDM filter with the variable  $r(k)$  to detect the corrupted pixels which are on the thin line component [12]. The variable  $r(k)$  is defined as follows:

$$r(k) = \min \left\{ |X_{ij} \otimes K_t| \mid t = 1, 2, 3, 4 \right\} \quad (11)$$

where  $K_t$  is the  $t$ th convolution kernel and  $\otimes$  represents the convolution operator. Four different absolute values are obtained that the filter window  $X_{ij}$  is operated with 4 different convolution kernels, respectively. The minimum of four values is selected for impulse noise detection [12]. Four different convolution kernels are shown in Fig. 3. In convolution operation, each pixel value in the filter window  $X_{ij}$  is multiplied by the corresponding weight. The result for convolution operation is the summary of all operation results. That is, it is the summary for each pixel in the filter window  $X_{ij}$  multiplied by the weight of the corresponding position in the  $t$ th convolution kernel.

0	0	0	0	0
0	0	0	0	0
-1	-1	4	-1	-1
0	0	0	0	0
0	0	0	0	0

0	0	-1	0	0
0	0	-1	0	0
0	0	4	0	0
0	0	-1	0	0
0	0	-1	0	0

-1	0	0	0	0
0	-1	0	0	0
0	0	4	0	0
0	0	0	-1	0
0	0	0	0	-1

0	0	0	0	-1
0	0	0	-1	0
0	0	4	0	0
0	-1	0	0	0
-1	0	0	0	0

Fig. 3. four  $5 \times 5$  convolution kernels

The variable  $r(k)$  utilized to detect impulse noise has 3 following significations:

1. Large  $r(k)$  value indicates all of 4 convolution products are very large. Thus,  $x(k)$  is considered as an impulse noise.
2. Small  $r(k)$  value denotes all of 4 convolution products are very small or close to zero. Hence,  $x(k)$  is regarded as a noise-free pixel.
3. Because one of 4 convolution products is very small (or close to zero) and the others are possibly large, the variable  $r(k)$  will be small when the central pixel  $x(k)$  is on the edge or the thin line component in the filter window  $X_{ij}$ .

According to aforementioned  $r(k)$ , if the input pixel  $x(k)$  is considered as noise-corrupted, the minimum of four convolution products is relatively large. The  $r(k)$  for the noise-free input pixel  $x(k)$  which is on the edge or the thin line component will be relatively small. Here, the threshold  $T_r$  will be employed to determine whether the pixel  $x(k)$  is corrupted or not.

#### Detector 8: the noise detection employing the SD-ROM filter

The variable  $\mathbf{h}(k)$  is defined as follows [13]:

$\mathbf{h}(k)$  is a one-dimensional vector that is obtained from the filter window  $x(k)$  without  $x_0(k)$ , shown by

$$\mathbf{h}(k) = (x_{-n}(k), \dots, x_{-1}(k), x_1(k), \dots, x_n(k)) \quad (12)$$

The elements in  $\mathbf{h}(k)$  are sorted by ascending order as follows:

$$\mathbf{h}'(k) = \{h'_{(1)}(k), h'_{(2)}(k), \dots, h'_{(2n)}(k)\} \quad (13)$$

where  $h'_{(1)}(k) \leq h'_{(2)}(k) \leq \dots \leq h'_{(2n)}(k)$ . Subsequently, the rank ordered mean (ROM) is defined as follows:

$$m(k) = (h'_{(2n/2)}(k) + h'_{((2n/2)+1)}(k))/2, \quad (14)$$

where  $m(k)$  is the value for the ROM. Also, the rank ordered difference is formulated by

$$\mathbf{z}(k) = (z_1(k), z_2(k), z_3(k), z_4(k)), \quad (15)$$

where

$$z_i(k) = \begin{cases} h'_{(i)}(k) - x(k), & \text{if } x(k) \leq m(k) \\ x(k) - h'_{(2n-i)}(k), & \text{if } x(k) > m(k) \end{cases} \quad \text{for } i=1, \dots, 4. \quad (16)$$

## Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

$\mathbf{z}(k)$  is employed to determine the input pixel  $x(k)$  whether it is corrupted or not. For instance, if  $z_1(k)$  is positive, it means that the input pixel  $x(k)$  is larger in the filter window  $\mathbf{x}(k)$ . If  $z_1(k)$  is greater than the threshold  $T_{z_1}$ , the input pixel  $x(k)$  is considered as noise-corrupted. Similarly,  $z_2(k)$ ,  $z_3(k)$  and  $z_4(k)$  can also provide the determination for noise detection. If each inequality in the following formula satisfies, the input pixel  $x(k)$  is regarded as noise-corrupted.

$$z_i(k) > T_{z_i}, \quad i=1,\dots,4, \quad (17)$$

where  $T_{Z_1}, T_{Z_2}, T_{Z_3}, T_{Z_4}$  are the thresholds and  $T_{Z_1} < T_{Z_2} < T_{Z_3} < T_{Z_4}$ .

**Detector 9:** the difference between the central pixel value and the neighbored pixels' values after sorted

In [14], Aizenberg proposed a noise detection algorithm about the variable  $e(k)$  to detect the corrupted pixels. The variable  $e(k)$  is defined as follows: (differential rank impulse detector, DRID)

$$e(k) = \begin{cases} |x(k) - \mathbf{x}'_{(a-1)}(k)|, & \text{if } a > (n+1) \\ |x(k) - \mathbf{x}'_{(a+1)}(k)|, & \text{if } a < (n+1) \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

The filter window  $\mathbf{x}(k)$  is sized with  $2n+1$ . After the elements in the filter window  $\mathbf{x}(k)$  being sorted, a new filter window  $\mathbf{x}'(k)$  is obtained. The rank for the input pixel  $x(k)$  in the new filter window  $\mathbf{x}'(k)$  is  $a$ .  $\mathbf{x}'_{(a-1)}(k)$  and  $\mathbf{x}'_{(a+1)}(k)$  represent the  $(a-1)$ th and the  $(a+1)$ th pixel's values in the new filter window  $\mathbf{x}'(k)$ , respectively. Similarly, large  $e(k)$  value which is larger than the threshold  $T_e$  represents that the input pixel  $x(k)$  is more relatively corrupted by impulse noise [12].

**Detector 10:** the noise detection employing the adaptive center-weighted median (ACWM) filter

The ACWM is a modified center-weighted median (CWM) filter [5]. It employs the difference between the input pixel  $x(k)$  and the output of the CWM filter to detect impulse noise.

$$o_i = |x(k) - c^s(k)| = |x(k) - c^{2i+1}(k)|, \quad (19)$$

where  $i = 0, 1, \dots, n-1$ , and  $c^i(k)$  means as Eq. (8).

If any inequality satisfies in the following equation, the input pixel  $x(k)$  is determined as an impulse noise [17].

$$o_i(k) > T_{o_i}, \quad i = 0, \dots, 3, \quad (20)$$

where  $T_{o_i}$  is the threshold. The corresponding threshold is obtained from the following equation:

$$T_{o_i} = \sigma \cdot \text{MAD} + \delta_i, \quad (21)$$

where  $\sigma$  and  $\delta_i$  are predefined parameters, and MAD represents the median of the absolute deviations from the median, defined as follows:

$$\text{MAD} = \text{median}(|x_{-n}(k) - c^1(k)|, \dots, |x_0(k) - c^1(k)|, \dots, |x_n(k) - c^1(k)|) \quad (22)$$

where the filter window  $X_{ij}$  is sized with  $2n+1$  and  $c^1(k)$  means as Eq. (8).

Each one of 10 noise detectors has its own decision threshold. Several tests by implementation must have been done before the optimal thresholds are obtained. The thresholds for 10 noise detectors will absolutely affect the accuracy rate in noise detection. Therefore, the optimal thresholds for training the DT with the PSO algorithm will be obtained. Also, the accuracy rate which the DT estimates is considered as the fitness for the PSO.

#### 4.3. The Training Method for the IND in the MDP Filter

Each one of 10 aforementioned detectors has its own threshold in the hybrid noise detector. The threshold in each detector usually affects the accuracy in noise detection for each noise detector. The optimal thresholds are determined only after many repeated experiments. Thus, a systematic method needs to be employed to obtain a set of optimal thresholds. Each noise detection algorithm has its strengths, weaknesses, and reciprocal characteristics. Therefore, the MDP filter adopts the DT to integrate each detector's characteristics and reciprocal relationships, and utilizes the PSO algorithm to determine the required optimal thresholds for the hybrid detection algorithm. Doing so can construct an adaptive hybrid noise detector to improve the detection rate and achieve better performance for image restoration.

The MDP filter adopts the PSO to find out the optimal thresholds for training the DT. Subsequently, it employs the error rate of the DT classification as the fitness of the PSO. Finally, it outputs a set of thresholds  $T$  determined by the PSO algorithm and a set of DT impulse noise detectors.

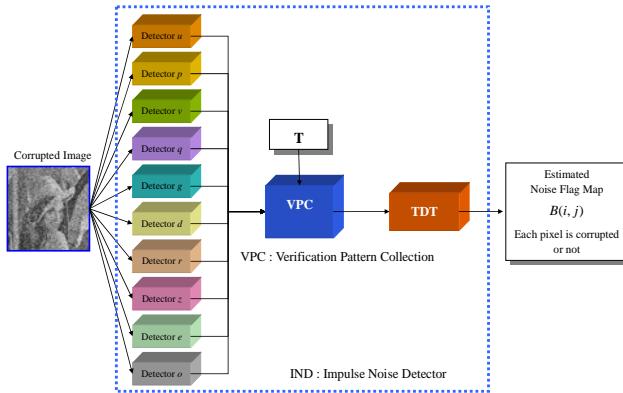
#### 4.4. The Structure for the IND

Fig. 4 shows the structure for the IND.  $\mathbf{T} = (T_u, T_p, T_v, T_q, T_g, T_d, T_r, T_{z_1}, T_{z_2}, T_{z_3}, T_{z_4}, T_e, \sigma, \delta_1, \delta_2, \delta_3, \delta_4)$ , where  $T_u, T_p, \dots, \sigma, \delta_1, \delta_2, \delta_3, \delta_4$  are the thresholds for Detector 1, Detector 2, ..., Detector 10, respectively. They are the optimal thresholds for the noise detection component which are found by the PSO algorithm.

The operation procedure is that the input image  $X$  employs the PSO to determine the optimal thresholds for noise judgment and then calculates the

## Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

judgment results for 10 detectors. Subsequently, the detection results of 10 noise detectors are fed into the trained DT (TDT) to judge whether the input pixel  $x(k)$  is corrupted and the MDP filter will output the binary noise flag map  $B(i, j)$ .  $B(i, j)$  clearly indicates that each pixel in the image is corrupted or not.



**Fig. 4.** The structure for the MDP noise detector

### 4.5. The MF, ROM, and PNOM Filters

In the MDP filter, after the noise detection procedure is finished, the binary flag map  $B(i, j)$  indicates the locations where the pixels are corrupted by impulse noise. When the input pixel  $x(k)$  is corrupted it will be replaced with the outputs of the MF, the ROM, and the PNOM filters, respectively. The MF filter outputs the median value of all sorted pixels in the filter window  $x(k)$ . The ROM filter calculates the average of the fourth and the fifth pixels for all sorted pixels without the central pixel in the original filter window  $x(k)$ . The PNOM filter sorts the noise-free pixels in the filter window  $x(k)$  by ascending order. They will be indicated as one dimensional vector as follows:

$$\mathbf{f}(k) = \{f_{(1)}(k), f_{(2)}(k), \dots, f_{(C)}(k)\}, \quad (23)$$

where  $f_{(1)}(k) \leq f_{(2)}(k) \leq \dots \leq f_{(C)}(k)$  and  $C$  represents the number of the noise-free pixels in the filter window  $x(k)$ .

Here, the algorithm for the PNOM filter is described as follows:

Step 1. Input the corrupted image  $X'$  and the noise detector, output the binary noise flag map  $B(i, j)$ .  $B(i, j) = 1$  indicates the central pixel  $x(k)$  is noise-corrupted.

Step 2. Slide the filter window  $x(k)$  over the corrupted image  $X'$  from left to right and top to bottom.

Step 2.1 Count the number  $C$  of the noise-free pixels in the filter window  $x(k)$  for each input corrupted pixel  $x(k)$ .

Step 2.2 If  $C > 0$ ,  $x(k) = \text{median}(\text{sort}(x(k)) \text{ excluding the corrupted pixels})$ ,  
 $B(i, j) = 0$ ,  $C = 0$ ,  
Else  $x(k)$  is kept unchanged.  
Step 3 Extend the filter window  $x(k)$  with size  $5 \times 5$ , perform Step 2 through  
Step 2.2  
Step 4 Output the restored image  $\hat{x}$ .

## 5. Experimental Results

Fig. 5 depicts that 10 different images which are gray-level images with size  $256 \times 256$  are adopted for the experiments in this work. They respectively represent ‘Airplane’, ‘Baboon’, ‘Barbara’, ‘Bridge’, ‘Butterfly’, ‘Couple’, ‘Fishingboat’, ‘Goldhill’, ‘Lena’ and ‘Peppers’.



**Fig. 5.** The images adopted for evaluation in the experiments

### 5.1. The Training and the Testing Methods Adopted in the Experiments

In order to evaluate the performance of impulse noise detector and image restoration in the MDP image filter, the training patterns are collected from the images at 20% noise ratio. In the training and the testing methods,  $P \in I_P$ , where  $P$  indicates the noise ratio,  $I_P \in \{2\%, 4\%, \dots, 18\%, 22\%, \dots, 30\%\}$ . The testing patterns are collected from the same images at 2~30% different noise ratios excluding 20%.

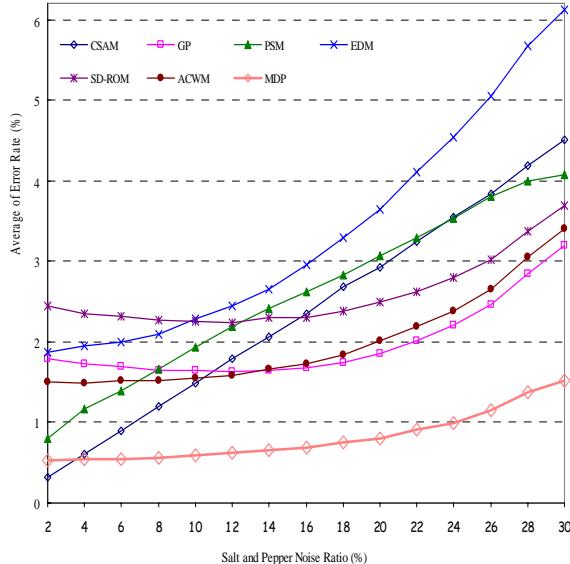
### 5.2. The Evaluation for the MDP Image Filter

To evaluate the performance for the MDP filter, the noise detection performance needs to be examined and its detector will be compared with various noise detectors applied on different images at different noise ratios. Also, the error rate in noise detection for the noise detectors is considered as

## Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and Particle Swarm Optimization for Image Restoration

the evaluating criterion. In the experiments, the existing well-known noise detection algorithms include: the CSAM [18], the GP [15], the SD-ROM [13], the PSM [4], the EDM [12], the ACWM [5] algorithms, where the PSM, the SD-ROM, and the ACWM algorithms employ the filter window with size  $3 \times 3$ , the EDM algorithm utilizes  $5 \times 5$  filter window, both the CSAM and the GP algorithms use the filter windows with size  $3 \times 3$  and  $5 \times 5$ . In the quality evaluation of image restoration, two other filters, the MF [1] and the TSM [3] filters, are also employed.

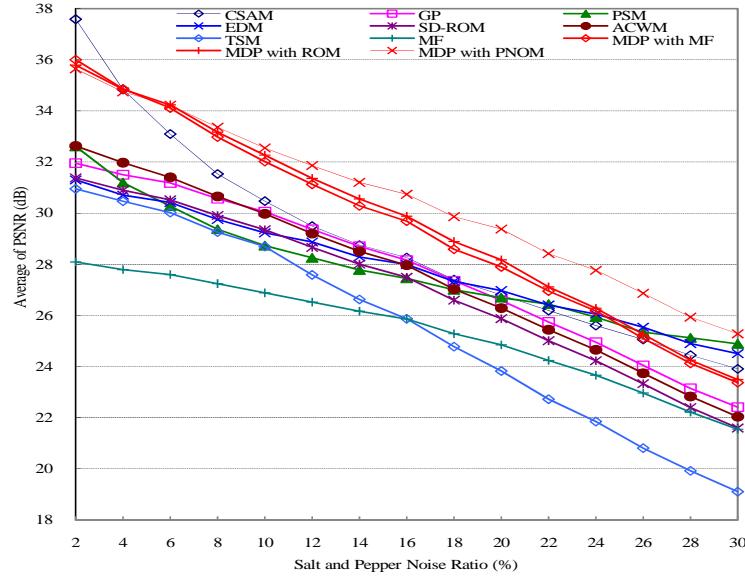
### 5.3. The Performance Evaluation for the MDP Filter Applied on the Images Corrupted by Salt and Pepper Impulse Noise



**Fig. 6.** The comparison for the average error rate in noise detection for different methods in 10 different images corrupted by the salt and pepper impulse noise at various noise ratios

Fig. 6 shows the quantitative comparison for the average error rate in noise detection for different detectors while 10 different images corrupted by salt and pepper impulse noise at various noise ratios. Fig. 7 depicts the quantitative comparison for the average PSNR in image restoration for different methods while 10 different images corrupted by salt and pepper impulse noise at various noise ratios. Fig. 8 illustrates the visual comparison

for different methods in image restoration when the image ‘Butterfly’ is corrupted by salt and pepper impulse noise at 24% noise ratio. Fig. 9 demonstrates the visual partial enlargement for the image ‘Butterfly’.

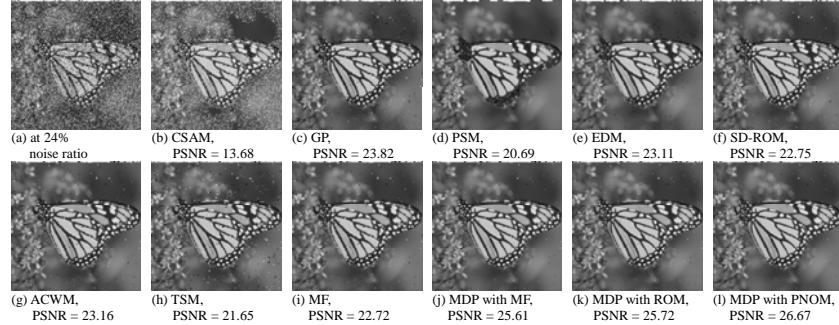


**Fig. 7.** The comparison for the average PSNR (dB) in image restoration for different methods in 10 different images corrupted by salt and pepper impulse noise at various noise ratios

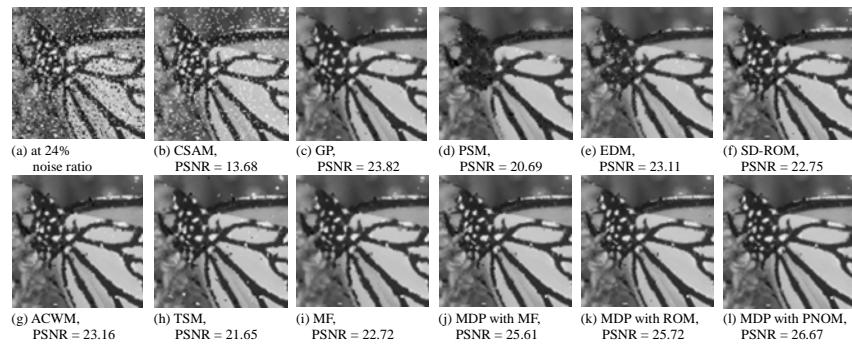
#### 5.4. The Performance Evaluation for the MDP Filter Applied on the Images Corrupted by Fixed-length Impulse Noise

Fig. 10 shows the quantitative comparison for the average error rate in noise detection for different detectors while 10 different images corrupted by fixed-length impulse noise at various noise ratios. Fig. 11 depicts the quantitative comparison for the average PSNR in image restoration for different methods while 10 different images corrupted by fixed-length impulse noise at various noise ratios. Fig. 12 illustrates the visual comparison for different methods in image restoration when the image ‘Lena’ is corrupted by fixed-length impulse noise at 18% noise ratio. Fig. 13 demonstrates the visual partial enlargement for the image ‘Lena’.

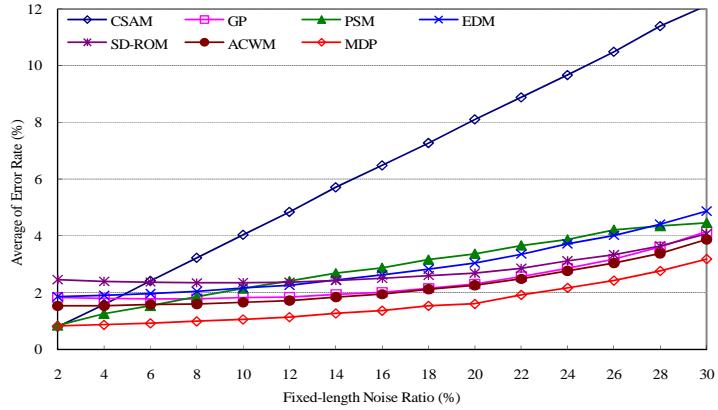
Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and  
Particle Swarm Optimization for Image Restoration



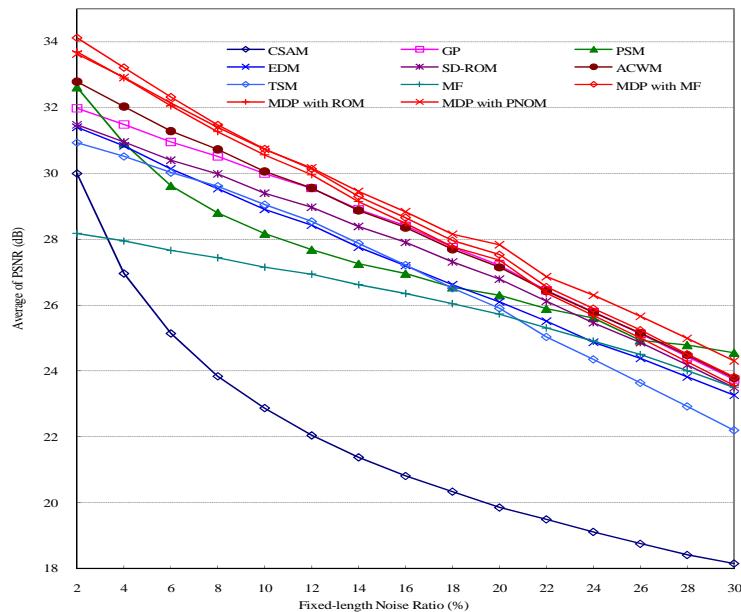
**Fig. 8.** The comparison for different methods in image restoration for the image 'Butterfly' is corrupted by salt and pepper impulse noise at 24% noise ratio



**Fig. 9.** The comparison for the partial enlargement using different methods in image restoration for the image 'Butterfly' corrupted by salt and pepper impulse noise at 24% noise ratio

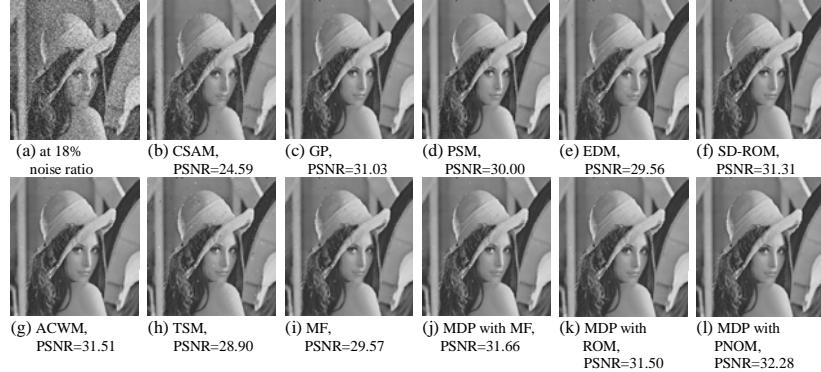


**Fig. 10.** The comparison for the average error rate in noise detection for different methods in 10 different images corrupted by fixed-length impulse noise at various noise ratios

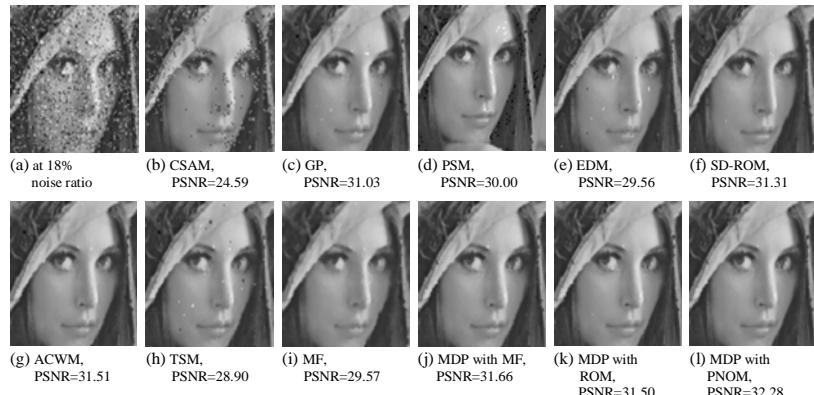


**Fig. 11.** The comparison for the average PSNR (dB) in image restoration for different methods in 10 different images corrupted by fixed-length impulse noise at various noise ratios

Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and  
Particle Swarm Optimization for Image Restoration



**Fig. 12.** The comparison for different methods in image restoration in the image ‘Lena’ is corrupted by fixed-length impulse noise at 18% noise ratio



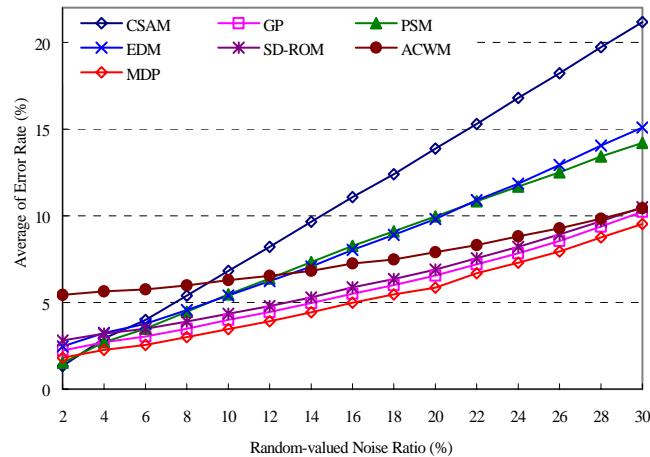
**Fig. 13.** The comparison for the partial enlargement for different methods in image restoration for the image ‘Lena’ corrupted by fixed-length impulse noise at 18% noise ratio

The important observation is proposed in Figs. 6 and 7. Although the CSAM filter can effectively detect the locations where the input pixels are corrupted by salt and pepper impulse noise and restore the corrupted images, it can not achieve good results for image restoration when the images are corrupted by fixed-length impulse noise. The reason is that the pixels will be nearly considered as noise-corrupted by the CSAM filter when their gray-level values are 0 or 255. Therefore, although the CSAM filter can achieve good performance in image restoration for filtering salt and pepper impulse noise, it can not obtain good results for fixed-length impulse noise.

### 5.5. The Performance Evaluation for the MDP Filter When the Images Are Corrupted by Random-valued Impulse Noise

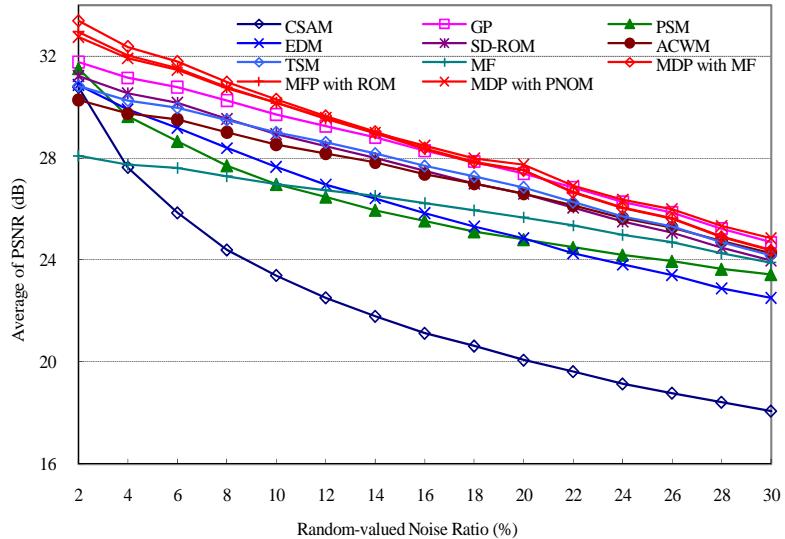
Fig. 14 shows the quantitative comparison for the average error rate in noise detection for different detectors while 10 different images corrupted by random-valued impulse noise at various noise ratios. Fig. 15 depicts the quantitative comparison for the average PSNR in image restoration for different methods while 10 different images corrupted by random-valued impulse noise at various noise ratios. Fig. 16 illustrates the visual comparison for different methods in image restoration when the image ‘Fishingboat’ is corrupted by random-valued impulse noise at 10% noise ratio. Fig. 17 demonstrates the visual partial enlargement for the image ‘Fishingboat’.

An important observation is presented again in Figs. 6 and 7. Although the CSAM filter can effectively detect the corrupted pixels caused by salt and pepper impulse noise and restore the corrupted images, it cannot also achieve good results in image restoration for fixed-length or random-valued impulse noise.

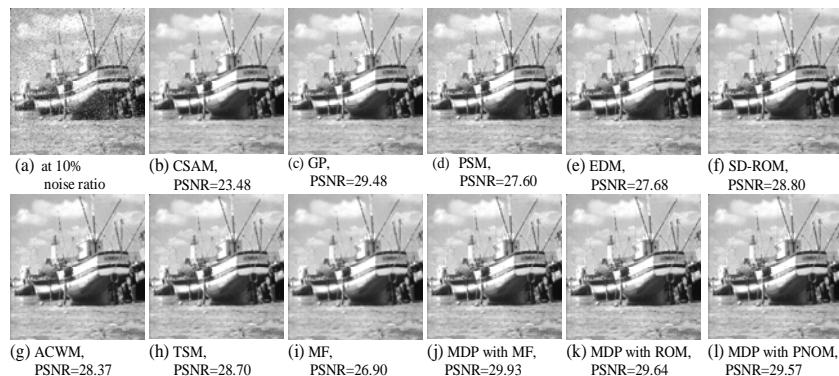


**Fig. 14.** The comparison for the average error rate in noise detection for different methods in 10 different images corrupted by random-valued impulse noise at various noise ratios

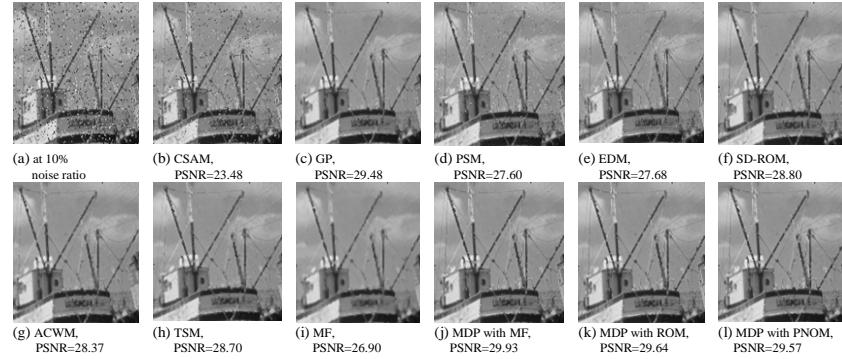
Design of Median-type Filters with an Impulse Noise Detector Using Decision Tree and  
Particle Swarm Optimization for Image Restoration



**Fig. 15.** The comparison for the average PSNR (dB) in image restoration for different methods in 10 different images corrupted by random-valued impulse noise at various noise ratios



**Fig. 16.** The comparison for different methods in image restoration in the image 'Fishingboat' corrupted by random-valued impulse noise at 10% noise ratio



**Fig. 17.** The comparison for the partial enlargement for different methods in image restoration for the image ‘Fishingboat’ corrupted by random-valued impulse noise at 10% noise ratio

## 6. Conclusion

This paper has proposed a novel image filter, called the MDP filter, which is used for the recovery of the corrupted pixels by impulsive noises in gray-level images. The IND is an adaptive hybrid noise detector which is constructed by integrating 10 impulse noise detectors based on the DT and the PSO. In the MDP filter, the high performance IND is employed to powerfully detect impulse noises and the modified MFs effectively restore for the corrupted images. Subsequently, the restoring process in the MDP filter respectively utilizes the MF, the ROM, and the PNOM filters to restore the corrupted pixels. The computational complexity of the proposed filter is high. It shall be improved with a modified algorithm in the future work. Experimental results demonstrate that the MDP filter can effectively restore the gray-level images corrupted by impulse noises and outperform the existing well-known methods.

**Acknowledgment.** This work is partially supported by the National Science Council of Taiwan, R.O.C., for its financially supporting the research under Contract No. NSC 97-2221-E-150-070. The authors also gratefully acknowledge the helpful comments of the reviewers for promoting the quality of the paper.

## References

1. Astola, J., Kuosmanen, P.: Fundamentals of Nonlinear Digital Filtering: CRC Press. Boca Raton, FL. (1997)
2. Yin, L., Yang, R., Gabbouj, M., Neuvo, Y.: Weighted Median Filters: a Tutorial. *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, Vol. 43, No. 3, 157-192, Mar. (1996)
3. Chen, T., Ma, K. K., Chen, L. H.: Tri-state Median Filter for Image Denoising. *IEEE Trans. Image Process.*, Vol. 8, 1834-1838, Dec. (1999)
4. Zhou, W., David, Z.: Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, Vol. 46, No. 1, 78-80, Jan. (1999)
5. Chen, T., Wu, H. R.: Adaptive Impulse Detection Using Center-weighted Median Filters. *IEEE Signal Processing Letters*, Vol. 8, 1-3. (2001)
6. Chen, T., Wu, H. R.: Impulse Noise Removal by Multi-state Median Filtering. In Proc. ICASSP'2000, Istanbul, Turkey, 2183-2186. (2000)
7. Forouzan, A. R., Araabi, B. N.: Iterative Median Filtering for Restoration of Images with Impulsive Noise. *IEEE Proceedings of International Conference, Electronics,Circuits and Systems*, Vol. 1, 232-235, Dec. (2003)
8. Dong, Y., Xu, S.: A New Directional Weighted Median Filter for Removal of Random-valued Impulse Noise. *IEEE Signal Processing Letters*, Vol. 14, 193-196, Mar. (2007)
9. Awad, A. S., Man, H.: High Performance Detection Filter for Impulse Noise Removal in Images. *Electronics Letters*, Vol.44, 192-194, Jan. (2008)
10. Lin, T. C., Yu, P. T.: Thresholding Noise-free Ordered Mean Filter Based on Dempster-Shafer Theory for Image Restoration. *IEEE Trans. on Circuits and Systems*, Vol. 53, No. 5, 1057-1064, May. (2006)
11. Ko, S. J., Lee, Y. H.: Center Weighted Median Filters and Their Applications to Image Enhancement. *IEEE Trans. Circuits System*, Vol. 38, No. 9, 984-993, Sep. (1991)
12. Zhang, S., Karim, M. A.: A New Impulse Detector for Switching Median Filters. *IEEE Signal Processing Lett.*, Vol. 9, 360-363, Nov. (2002)
13. Abreu, E., Lightstone, M., Mitra, S. K., Arakawa, K.: A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images. *IEEE Trans. Image Processing*, Vol. 5, 1012-1025, June. (1996)
14. Aizenberg, I., Butakoff, C.: Effective Impulse Detector Based on Rank Order Criteria. *IEEE Signal Processing Lett.*, Vol. 11, 363-366, Mar. (2004)
15. Petrovic, N. I., Crnojevic, V.: Universal Impulse Noise Filter Based on Genetic Programming. *IEEE Trans. on Image Processing*, Vol. 17, Issue 7, 1109-1120, Jul. (2008)
16. Tan, P. N., Steinbach, M., Kumar, V.: Introduction to Data Mining: Addison-Wesley. (2006)
17. You, P.S., Hsieh, Y. C., Huang, C. M.: A Particle Swarm Optimization Based Algorithm to the Internet Subscription Problem. *Expert Systems with Applications*, Vol. 36, Issue 3, 7093-7098, Apr. (2009)
18. Pok, G., Liu, J. C., Nair, A. S.: Selective Removal of Impulse Noise Based on Homogeneity Level Information. *IEEE Trans. on Image Processing*, Vol. 12, No. 1, 85-92, Jan. (1999)

Bae-Muu Chang, Hung-Hsu Tsai, Xuan-Ping Lin, and Pao-Ta Yu

**Bae-Muu Chang** received the BS degree in computer science from Tamkang University, Taipei, Taiwan, in 1977, the MS degree in computer science from New York Institute of Technology, New York, USA, in 1989, and the PhD degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 2009. Since 1990, he has been with the Department of Information Management at Chienkuo Technology University, Changhua, Taiwan, where he is currently an associate professor. His research interests include neural networks, fuzzy systems, character recognition, intelligent filter design, intelligent networks, and e-Learning.

**Hung-Hsu Tsai** received the BS and the MS degrees in applied mathematics from National Chung Hsing University, Taichung, Taiwan, in 1986 and 1988, respectively, and the PhD degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 1999. Currently, he is a professor at Department of Information Management, National Formosa University, Huwei, Yulin, Taiwan. He has worked in industry for the SYSTEK Corporation, and in academia for Nanhua University, Chiayi, Taiwan. He is an honorary member of the Phi Tau Phi Scholastic Honor Society. He has been selected and included in the 9th edition of Who's Who in Science and Engineering which has been published in 2006. He serves as a technical reviewer for various scientific journals and numerous international conferences. His research interests include computational intelligence, machine learning, support vector machines, multimedia security, digital watermarking, intelligent filter design, content-based multimedia retrieval, data mining, e-Learning and system integration with web services.

**Pao-Ta Yu** received the BS degree in mathematics from National Taiwan Normal University, Taipei, Taiwan, in 1979, the MS degree in computer science from National Taiwan University, Taipei, Taiwan, in 1985, and the PhD degree in electrical engineering from Purdue University, West Lafayette, Indiana, in 1989. Since 1990, he has been with the Department of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan, where he is currently a professor. His research interests include neural networks and fuzzy systems, nonlinear filter design, intelligent networks, XML technology and e-Learning.

**Xuan-Ping Lin** respectively received the BS degree and the MS degree in computer science & information engineering and information management from National Formosa University, Huwei, Yulin, Taiwan, in 2006 and 2009. He is currently an engineer at AU Optronics Corporation. His research interests include particle swarm optimization, support vector machine, data mining, and intelligent filter design.

*Received: April 05, 2009; Accepted: May 21, 2010.*