

# The Optimal Design of Weighted Order Statistics Filters by Using Support Vector Machines

Chih-Chia Yao and Pao-Ta Yu

*Department of Computer Science and Information Engineering, College of Engineering, National Chung Cheng University, Chia-yi 62107, Taiwan*

Received 10 January 2005; Revised 13 September 2005; Accepted 7 November 2005

Recommended for Publication by Moon Gi Kang

Support vector machines (SVMs), a classification algorithm for the machine learning community, have been shown to provide higher performance than traditional learning machines. In this paper, the technique of SVMs is introduced into the design of weighted order statistics (WOS) filters. WOS filters are highly effective, in processing digital signals, because they have a simple window structure. However, due to threshold decomposition and stacking property, the development of WOS filters cannot significantly improve both the design complexity and estimation error. This paper proposes a new designing technique which can improve the learning speed and reduce the complexity of designing WOS filters. This technique uses a dichotomous approach to reduce the Boolean functions from 255 levels to two levels, which are separated by an optimal hyperplane. Furthermore, the optimal hyperplane is gotten by using the technique of SVMs. Our proposed method approximates the optimal weighted order statistics filters more rapidly than the adaptive neural filters.

Copyright © 2006 Hindawi Publishing Corporation. All rights reserved.

## 1. INTRODUCTION

Support vector machines (SVMs), a classification algorithm for the machine learning community, have attracted much attention in recent years [1–5]. In many applications, SVMs have been shown to provide higher performance than traditional learning machines [6–8].

The principle of SVMs is based on approximating structural risk minimization. It shows that the generalization error is bounded by the sum of the training set error and a term dependent on the Vapnik-Chervonenkis dimension of the learning machines [2]. The idea of SVMs originates from finding an optimal separating hyperplane which separates the largest possible fraction of training set of each class of data while maximizing the distance from either class to the separating hyperplane. According to Vapnik [9], this hyperplane minimizes the risk of misclassifying not only the examples in the training set, but also the unseen examples of the test set.

SVMs performance versus traditional learning machines suggested that a redesign approach could overcome significant problems under study [10–15]. In this paper, a new dichotomous technique for designing WOS filter by SVMs is proposed. WOS filters are a special subset of stack filters, and

are used in a lot of applications including noise cancellation, image restoration, and texture analysis [16–21].

Each stack filter based on a positive Boolean function can be characterized by two properties—threshold decomposition and stacking property [11, 22]. The Boolean function on which each WOS filter is based is a threshold logic which needs an  $n$ -dimensional weight vector and a threshold value. The representation of WOS filters based on threshold decomposition involves  $K - 1$  Boolean functions while input data are decomposed into  $K - 1$  levels. Note that  $K$  is the number of gray levels of the input data. This architecture has been realized in multilayer neural networks [20]. However, based on stacking property, the boolean function can be reduced from  $K - 1$  levels to two levels without loss of accuracy.

Several research studies into WOS filters have also been proposed recently [23–27]. Due to threshold decomposition and stacking property, these studies cannot significantly improve the design complexity and estimation error of WOS filters. This task can be accomplished, however, when the concept of SVMs is involved to reduce the Boolean functions. This paper compares our algorithm with adaptive neural filters, first proposed by Yin et al. [20], approximating the solution of minimum estimation error. Yin et al. applied a backpropagation algorithm to develop adaptive neural filters

with sigmoidal neuron functions as their nonlinear threshold functions [20]. The learning process of adaptive neural filters has a long computational time since the learning structure is based on the architecture of threshold decomposition; that is, the learning data at each level of threshold decomposition must be manipulated. One contribution of this paper is to design an efficient algorithm for approximating an optimal WOS filter. In this algorithm, the total computational time is only  $2T$  (time units), whereas the adaptive neural filter has a computational time of  $255T$  (time units), given training data of 256 gray levels. Our experimental results are superior to those obtained using adaptive neural filters. We believe that the design methodology in our algorithm will reinvigorate research into stack filter, including morphological filters which have languished for a decade.

This paper is organized as follows. In Section 2, the basic concepts of SVMs, WOS filters, and adaptive neural filters are reviewed. In Section 3, the concept of dichotomous WOS filters is described. In Section 4, a fast algorithm for generating an optimal WOS filter by SVMs is proposed. Finally, some experimental results are presented in Section 5 and our conclusions are offered in Section 6.

## 2. BASIC CONCEPTS

This section reviews three concepts: the basic concept of SVMs, the definition of WOS filters with reference to both the multivalued domain and binary domain approaches, and finally adaptive neural filters proposed by Yin et al. [2, 20].

### 2.1. Linear support vector machines

Consider the training samples  $\{(x_i, y_i)\}_{i=1}^L$ , where  $x_i$  is the input pattern for the  $i$ th sample and  $y_i$  is the corresponding desired response;  $x_i \in R^m$  and  $y_i \in \{-1, 1\}$ . The objective is to define a separating hyperplane which divides the set of samples such that all the points with the same class are on the same sides of the hyperplane.

Let  $w_o$  and  $b_o$  denote the optimum values of the weight vector and bias, respectively. The optimal separating hyperplane, representing a multidimensional linear decision surface in the input space, is given by

$$w_o^T x + b_o = 0. \quad (1)$$

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the margin of separation is maximal. Then, the separating hyperplane  $w^T x + b = 0$  must satisfy the following constraints:

$$y_i(w^T x_i + b) > 0, \quad i = 1, 2, \dots, L. \quad (2)$$

Equation (2) can be redefined without losing accuracy,

$$y_i(w^T x_i + b) \geq 1, \quad i = 1, 2, \dots, L. \quad (3)$$

When the nonseparable case is considered, a slack variable  $\xi_i$  is introduced to measure the deviation of a data point from an ideal value which would yield pattern separability. Hence,

the constraint of (3) is modified to

$$\begin{aligned} y_i(w^T x_i + b) &\geq 1 - \xi_i, \quad i = 1, 2, \dots, L, \\ \xi_i &\geq 0. \end{aligned} \quad (4)$$

Two support hyperplanes  $w^T x_i + b = 1$  and  $w^T x_i + b = -1$ , which define the two borders of margin of separation, are specified on (4). According to (4), the optimal separating hyperplane is the maximal margin hyperplane with the geometric margin  $2/\|w\|$ . Hence, the optimal separating hyperplane is the one that satisfies (4) and minimizes the cost function,

$$\Phi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^L \xi_i. \quad (6)$$

The parameter  $C$  controls the tradeoff between the complexity of the machine and the number of nonseparable points. The parameter  $C$  is selected by the user. A larger  $C$  assigns a higher penalty to errors.

Since the cost function is a convex function, a Lagrange function can be used to minimize the constrained optimization problem:

$$\begin{aligned} L(w, b, \alpha) &= \frac{1}{2} w^T w + C \sum_{i=1}^L \xi_i - \sum_{i=1}^L \alpha_i [y_i(w^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^L \beta_i \xi_i, \end{aligned} \quad (7)$$

where  $\alpha_i, \beta_i, i = 1, 2, \dots, L$ , are the Lagrange multipliers.

Once the solution  $\alpha^o = (\alpha_1^o, \alpha_2^o, \dots, \alpha_L^o)$  of (7) has been found, the optimal weight vector is given by

$$w^o = \sum_{i=1}^L \alpha_i^o y_i x_i. \quad (8)$$

Classical Lagrangian duality enables the primal problem to be transformed to its dual problem. The dual problem of (7) is reformulated as

$$Q(\alpha) = \sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j y_i y_j x_i^T x_j, \quad (9)$$

with constraints

$$\sum_{i=1}^L \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, L. \quad (10)$$

### 2.2. Nonlinear support vector machines

Input data can be mapped onto an alternative, higher-dimensional space, called *feature space* through a replacement to improve the representation,

$$x_i \cdot x_j \rightarrow \varphi(x_i)^T \varphi(x_j). \quad (11)$$

The functional form of the mapping  $\varphi(\cdot)$  does not need to be known since it is implicitly defined by selected kernel function  $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ , such as polynomials, splines,

radial basis function networks, or multilayer perceptrons. A suitable choice of kernel can make the data separable in feature space despite being nonseparable in the original input space. For example, the XOR problem is nonseparable by a hyperplane in input space, but it can be separated in the feature space defined by the polynomial kernel,

$$K(x, x_i) = (x^T x_i + 1)^P. \quad (12)$$

When  $x_i$  is replaced by its mapping in the feature space  $\varphi(x_i)$ , (9) becomes

$$Q(\alpha) = \sum_{i=1}^L \alpha_i - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j y_i y_j K(x_i, x_j). \quad (13)$$

### 2.3. WOS filters

In the multivalued domain  $\{0, 1, \dots, K-1\}$ , the output of a WOS filter can be easily obtained by a sorting operation. Let the  $K$ -valued input sequence or signal be  $\vec{X} = (X_1, X_2, \dots, X_L)$  and let the  $K$ -valued output sequence be  $\vec{Y} = (Y_1, Y_2, \dots, Y_L)$ , where  $X_i, Y_i \in \{0, 1, \dots, K-1\}$ ,  $i \in \{1, 2, \dots, L\}$ . Then, the output  $Y_i = F_{\underline{W}}(\vec{X}_i)$  can be obtained according to the following equation, where  $\vec{X}_i = (X_{i-N}, \dots, X_i, \dots, X_{i+N})$  and  $F_{\underline{W}}(\cdot)$  denotes the filtering operation of the WOS filter associated with the corresponding vector  $\underline{W}$  consisting of weights and threshold:

$Y_i = F_{\underline{W}}(\vec{X}_i) =$  the  $t$ th largest value of the samples

$$\left\{ \overbrace{X_{i-N}, \dots, X_{i-N}}^{w_1 \text{ times}}, \overbrace{X_{i-N+1}, \dots, X_{i-N+1}}^{w_2 \text{ times}}, \dots, \overbrace{X_{i+N}, \dots, X_{i+N}}^{w_{2N+1} \text{ times}} \right\}, \quad (14)$$

where  $\underline{W} = [w_1, w_2, \dots, w_{2N+1}; t]^T$  and  $T$  denotes transpose. The terms  $w_1, w_2, \dots, w_{2N+1}$  and  $t$  are all nonnegative integers. Then, a necessary and sufficient condition for  $X_k$ ,  $i-N \leq k \leq i+N$ , being the output of a WOS filter, is

$$k = \min \left\{ j \mid \sum_{i=1}^j w_i \geq t \right\}. \quad (15)$$

The WOS filter is defined, using (15). In such a definition, the weights and threshold value need not be nonnegative integers. They can be any nonnegative real numbers [15, 28].

Using (15), the output  $f(\vec{x})$  of a WOS filter for a binary input vector  $\vec{x} = \{x_{i-N}, x_{i-N+1}, \dots, x_i, \dots, x_{i+N}\}$  is written as

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \sum_{j=i-N}^{i+N} w_j x_j \geq t, \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

The function  $f(\vec{x})$  is a special case of Boolean functions, and is called the threshold function. Since WOS filters have nonnegative weights and threshold, they are stack filters.

As a subclass of stack filters, WOS filters have representations in the threshold decomposition architecture. Assuming

that  $X_i \in \{0, 1, \dots, K-1\}$  for all  $i$ , it can be decomposed into  $K-1$  binary sequence  $\{X_i^m\}_{m=1}^{K-1}$  by thresholding. This thresholding operation is called  $T_m$  and is defined as

$$X_i^m = T_m(X_i) = U(X_i - m) = \begin{cases} 1 & \text{if } X_i \geq m, \\ 0 & \text{otherwise,} \end{cases} \quad (17)$$

where  $U(\cdot)$  is a unit step function;  $U(x) = 1$  if  $x \geq 0$  and  $U(x) = 0$  if  $x < 0$ . Note that

$$X_i = \sum_{m=1}^{K-1} T_m(X_i) = \sum_{m=1}^{K-1} X_i^m. \quad (18)$$

By using the threshold decomposition architecture, WOS filters can be implemented by threshold logic. That is, the output of WOS filters is defined as

$$Y_i = \sum_{m=1}^{K-1} U(\underline{W}^T \underline{X}_i^m), \quad i = 1, 2, \dots, L, \quad (19)$$

where  $\underline{X}_i^m = [X_{i-N}^m, X_{i-N+1}^m, \dots, X_i^m, \dots, X_{i+N}^m, -1]^T$ .

### 2.4. Adaptive neural filters

Let  $\vec{X} = (X_1, X_2, \dots, X_L)$  and  $\vec{Z} = (Z_1, Z_2, \dots, Z_L) \in \{0, 1, \dots, K-1\}^L$  be the input and the desired output of the adaptive neural filter, respectively. If  $X_i$  and  $Z_i$  are jointly stationary, then the MSE to be minimized is

$$\begin{aligned} J(\underline{W}) &= E \left[ (Z_i - F_{\underline{W}}(\vec{X}_i))^2 \right] \\ &= E \left[ \left( \sum_{n=1}^{K-1} (T_n(Z_i) - \sigma(\underline{W}^T \underline{X}_i^n)) \right)^2 \right]. \end{aligned} \quad (20)$$

Note that  $\sigma(x) = 1/(1 + e^{-x})$  is the sigmoid function instead of the unit step function  $U(\cdot)$ . Analogous to the backpropagation algorithm, the optimal adaptive neural filter can be derived by applying the following update rule [20]:

$$\underline{W} \leftarrow \underline{W} + \mu \Delta \underline{W} = \underline{W} + 2\mu (Z_i - F_{\underline{W}}(\vec{X}_i)) \sum_{n=1}^{K-1} s_i^n (1 - s_i^n) \underline{X}_i^n, \quad (21)$$

where  $\mu$  is a learning rate and  $s_i^n = \sigma(\underline{W}^T \underline{X}_i^n) \in [0, 1]$ , that is,  $s_i^n$  is the approximate output of  $F_{\underline{W}}(\vec{X}_i)$  at level  $n$ . The learning process can be repeated from  $i = 1$  to  $L$  or with more iterations.

These filters use a sigmoid function as a neuron activation function, which can approximate linear functions and unit step functions. Therefore, they can approximate both FIR filters and WOS filters. However, the above algorithm takes much computational time to sum up the  $(K-1)$  binary signals, and it is difficult to understand the correlated behaviors among signals. This motivates the development of another approach which is presented in the next section to reduce the computational cost and clarify the correlated behaviors of signals with the viewpoint of support vector.

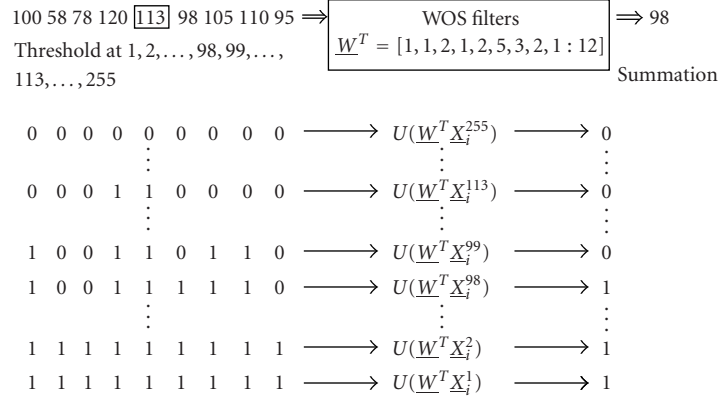


FIGURE 1: The filtering behavior of WOS filters when  $X_i = 113$ .

### 3. A NEW DICHOTOMOUS TECHNIQUE FOR DESIGNING WOS FILTERS

This section proposes a new approach which adopts the concept of dichotomy and reduces Boolean functions with  $K - 1$  levels into Boolean functions with only two levels, thus saving considerable computational time.

Recall the definition of WOS filters from the previous section. Let  $\underline{X}_i^n = [x_{i-N}, x_{i-N+1}, \dots, x_i, \dots, x_{i+N}, -1]^T$ ;  $x_i = 1$  if  $X_i \geq n$  and  $x_i = 0$  if  $X_i < n$ ; and  $\underline{W}^T = [w_{i-N}, w_{i-N+1}, \dots, w_i, \dots, w_{i+N}, t]$ . Using (16), the output of a WOS filter for a binary input vector  $(x_{i-N}, x_{i-N+1}, \dots, x_i, \dots, x_{i+N})$  is written as

$$U(\underline{W}^T \underline{X}_i^n) = \begin{cases} 1 & \text{if } \sum_{k=i-N}^{i+N} w_k x_k \geq t, \\ 0 & \text{if } \sum_{k=i-N}^{i+N} w_k x_k < t. \end{cases} \quad (22)$$

In the multivalued domain  $\{0, 1, \dots, K - 1\}$ , the architecture of threshold decomposition has  $K - 1$  unit step functions. Suppose the output value of  $Y_i$  is  $m$ , and then  $Y_i$  can be decomposed as (23) by threshold decomposition,

$$Y_i = m \Rightarrow \text{decomposition of } Y_i = \left\{ \overbrace{1, \dots, 1}^{m \text{ times}}, \overbrace{0, \dots, 0}^{K-1-m \text{ times}} \right\}. \quad (23)$$

Besides,  $\underline{X}_i$  is also decomposed into  $K - 1$  binary vectors  $\underline{X}_i^1, \underline{X}_i^2, \dots, \underline{X}_i^{K-1}$ . Then, those  $K - 1$  outputs of the unit step function are listed as follows:  $U(\underline{W}^T \underline{X}_i^1), U(\underline{W}^T \underline{X}_i^2), \dots, U(\underline{W}^T \underline{X}_i^{K-1})$ . According to the stacking property [22],

$$\underline{X}_i^1 \geq \underline{X}_i^2 \geq \dots \geq \underline{X}_i^{K-1} \Rightarrow U(\underline{W}^T \underline{X}_i^1) \geq U(\underline{W}^T \underline{X}_i^2) \geq \dots \geq U(\underline{W}^T \underline{X}_i^{K-1}). \quad (24)$$

It implies  $U(\underline{W}^T \underline{X}_i^1) = 1, U(\underline{W}^T \underline{X}_i^2) = 1, \dots, U(\underline{W}^T \underline{X}_i^m) = 1, U(\underline{W}^T \underline{X}_i^{m+1}) = 0, \dots, U(\underline{W}^T \underline{X}_i^{K-1}) = 0$ . Then, two conclusions are formulated: (a) for all  $j \leq m$ ,  $U(\underline{W}^T \underline{X}_i^j) = 1$  and (b) for all  $j \geq m + 1$ ,  $U(\underline{W}^T \underline{X}_i^j) = 0$ . Consequently, if the

output  $Y_i$  equals  $m$ , the definition of the WOS filters can be rewritten as

$$Y_i = m = \sum_{n=1}^{K-1} U(\underline{W}^T \underline{X}_i^n) = \sum_{n=1}^m U(\underline{W}^T \underline{X}_i^n). \quad (25)$$

Figure 1 illustrates this concept. It shows the filtering behavior of a window width  $3 \times 3$  WOS filter, based on the architecture of threshold decomposition. The data in the upper left are input signals and the data in the upper right are the output, after WOS filtering. The 256-valued input signals are decomposed into a set of 255 binary signals. After thresholding, each binary signal is independently processed according to (22). Finally, the outputs of the unit step function are summed.

In Figure 1, the threshold value  $t$  is 12; this means that the 12th largest value from the set  $\{100, 58, 78, 78, 120, 113, 113, 98, 98, 98, 98, 98, 105, 105, 105, 110, 110, 95\}$  is chosen. The physical output of the WOS filter is then 98. Figure 1 indicates that

- (i) for all  $n \leq 98$ , where  $n$  is an integer,  $\underline{X}_i^n \geq \underline{X}_i^{98}$  and  $\underline{W}^T \underline{X}_i^n \geq \underline{W}^T \underline{X}_i^{98}$ . When  $\underline{W}^T \underline{X}_i^{98} = 1$ , then  $\underline{W}^T \underline{X}_i^n$  must equal one;
- (ii) for all  $n \geq 99$ , where  $n$  is an integer,  $\underline{X}_i^n \leq \underline{X}_i^{99}$  and  $\underline{W}^T \underline{X}_i^n \leq \underline{W}^T \underline{X}_i^{99}$ . When  $\underline{W}^T \underline{X}_i^{99} = 0$ , then  $\underline{W}^T \underline{X}_i^n$  must equal zero.

In the supervised learning mode, if the desired output is  $m$ , then the goal in designing a WOS filter is to adjust the weight vector such that it satisfies  $U(\underline{W}^T \underline{X}_i^{m+1}) = 0$  and  $U(\underline{W}^T \underline{X}_i^m) = 1$ , implying that the input signal need not be considered at levels other than  $\underline{X}_i^{m+1}$  and  $\underline{X}_i^m$ . This concept is referred to as dichotomy.

Accordingly, the binary input signals  $\underline{X}_i^k, k \in \{1, 2, \dots, 255\}$ , are classified into 1-vector and 0-vector signals. The input signals  $\underline{X}_i^k$  are 1-vector if they satisfy  $U(\underline{W}^T \underline{X}_i^k) = 1$ . They are 0-vector if they satisfy  $U(\underline{W}^T \underline{X}_i^k) = 0$ . In vector space, these two classes are separated by an optimal hyperplane, which is bounded by  $\underline{W}^T \underline{X}_i^m \geq 0$  and  $\underline{W}^T \underline{X}_i^{m+1} < 0$ , when the output value is  $m$ . Hence, the vector  $\underline{X}_i^m$  is called the 1-support vector and the vector  $\underline{X}_i^{m+1}$  is called the 0-support

vector, because  $\underline{X}_i^m$  and  $\underline{X}_i^{m+1}$  are helpful in determining the optimal hyperplane.

#### 4. SUPPORT VECTOR MACHINES FOR DICHOTOMOUS WOS FILTERS

##### 4.1. Linear support vector machines for dichotomous WOS filters

In the above section, the new approach of designing WOS filter has reduced the Boolean functions with  $K - 1$  levels into two levels. In this section, the support vector machines are introduced on the design of dichotomous WOS filters. The new technique is illustrated as follows.

If the input vector is  $\underline{X}_i^n = [x_{i-N}, x_{i-N+1}, \dots, x_i, \dots, x_{i+N}, -1]^T$ ,  $n = 0, 1, \dots, 255$ , and the desired output is  $m$ , then an appropriate  $\underline{W}^T$  can be found, such that two constraints are satisfied:  $\underline{W}^T \underline{X}_i^m \geq 0$  and  $\underline{W}^T \underline{X}_i^{m+1} < 0$ . For increasing tolerance,  $\underline{W}^T \underline{X}_i^m \geq 0$  and  $\underline{W}^T \underline{X}_i^{m+1} < 0$  are redefined as follows:

$$\begin{aligned} \sum_{k=i-N}^{i+N} w_k x_{1k} - t &\geq 1, & x_{1k} \text{ is the } k\text{th component of } \underline{X}_i^m, \\ \sum_{k=i-N}^{i+N} w_k x_{2k} - t &\leq -1, & x_{2k} \text{ is the } k\text{th component of } \underline{X}_i^{m+1}. \end{aligned} \quad (26)$$

The corresponding outputs  $y_{1i}$ ,  $y_{2i}$  of (26) are  $y_{1i} = U(\underline{W}^T \underline{X}_i^m) = 1$  and  $y_{2i} = U(\underline{W}^T \underline{X}_i^{m+1}) = 0$ . When  $y_{1i}$  and  $y_{2i}$  are considered, (27) is obtained as follows:

$$\begin{aligned} (2y_{1i} - 1) \left( \sum_{k=i-N}^{i+N} w_k x_{1k} - t \right) &\geq 1, \\ x_{1k} \text{ is the } k\text{th component of } \underline{X}_i^m, \\ (2y_{2i} - 1) \left( \sum_{k=i-N}^{i+N} w_k x_{2k} - t \right) &\geq 1, \\ x_{2k} \text{ is the } k\text{th component of } \underline{X}_i^{m+1}. \end{aligned} \quad (27)$$

Let  $\vec{x}_{1i} = [x_{1(i-N)}, x_{1(i-N+1)}, \dots, x_{1i}, \dots, x_{1(i+N)}]^T$  and  $\vec{x}_{2i} = [x_{2(i-N)}, x_{2(i-N+1)}, \dots, x_{2i}, \dots, x_{2(i+N)}]^T$ . Then, (27) can be expressed in vector form as follows:

$$\begin{aligned} (2y_{1i} - 1)(w^T \vec{x}_{1i} - t) &\geq 1, \\ (2y_{2i} - 1)(w^T \vec{x}_{2i} - t) &\geq 1, \end{aligned} \quad (28)$$

where  $w^T = [w_{i-N}, w_{i-N+1}, \dots, w_i, \dots, w_{i+N}]$ . Equation (28) is similar to the constraint which is used in SVMs. Moreover, when misclassified data are considered, (28) is modified as follows:

$$\begin{aligned} (2y_{1i} - 1)(w^T \vec{x}_{1i} - t) + \xi_{1i} &\geq 1, \\ (2y_{2i} - 1)(w^T \vec{x}_{2i} - t) + \xi_{2i} &\geq 1, \\ \xi_{1i}, \xi_{2i} &\geq 0. \end{aligned} \quad (29)$$

Now, we formulate the optimal design of WOS filters as the following constrained optimization problem.

Given the training samples  $\{(\vec{X}_i, m_i)\}_{i=1}^L$ , find an optimal value of the weight vector  $w$  and threshold  $t$  such that they satisfy the constraints

$$(2y_{1i} - 1)(w^T \vec{x}_{1i} - t) + \xi_{1i} \geq 1, \quad \text{for } i = 1, 2, \dots, L, \quad (30)$$

$$(2y_{2i} - 1)(w^T \vec{x}_{2i} - t) + \xi_{2i} \geq 1, \quad \text{for } i = 1, 2, \dots, L, \quad (31)$$

$$w \geq 0, \quad (32)$$

$$t \geq 0, \quad (33)$$

$$\xi_{1i}, \xi_{2i} \geq 0, \quad \text{for } i = 1, 2, \dots, L, \quad (34)$$

and such that the weight vector  $w$  and the slack variables  $\xi_{1i}$ ,  $\xi_{2i}$  can minimize the cost function:

$$\Phi(w, \xi_1, \xi_2) = \frac{1}{2} w^T w + C \sum_{i=1}^L (\xi_{1i} + \xi_{2i}), \quad (35)$$

where  $C$  is a user-specified positive parameter and  $\vec{x}_{1i} = [X_{i-N}^{m_i}, X_{i-N+1}^{m_i}, \dots, X_i^{m_i}, \dots, X_{i+N}^{m_i}]^T$  and  $\vec{x}_{2i} = [X_{i-N}^{m_i+1}, X_{i-N+1}^{m_i+1}, \dots, X_i^{m_i+1}, \dots, X_{i+N}^{m_i+1}]^T$ . Note that the inequality constraint “ $w \geq 0$ ” defines that all elements in binary vector are equal to or bigger than 0. Since the cost function  $\phi(w, \xi_1, \xi_2)$  is a convex function of  $w$  and the constraints are linear in  $w$ , the above constrained optimization problem can be solved by using the method of Lagrange multipliers [29].

The Lagrangian function is introduced to solve the above problem. Let

$$\begin{aligned} L(w, t, \xi_1, \xi_2) &= \frac{1}{2} w^T w + C \sum_{i=1}^L (\xi_{1i} + \xi_{2i}) - \sum_{i=1}^L \alpha_i \\ &\quad \times [(2y_{1i} - 1)(w^T \vec{x}_{1i} - t) + \xi_{1i} - 1] \\ &\quad - \sum_{i=1}^L \beta_i [(2y_{2i} - 1)(w^T \vec{x}_{2i} - t) + \xi_{2i} - 1] \\ &\quad - \gamma^T w - \eta t - \sum_{i=1}^L \mu_{1i} \xi_{1i} - \sum_{i=1}^L \mu_{2i} \xi_{2i}, \end{aligned} \quad (36)$$

where the auxiliary nonnegative variables  $\alpha_i$ ,  $\beta_i$ ,  $\gamma$ ,  $\eta$ ,  $\mu_{1i}$ , and  $\mu_{2i}$  are called Lagrange multipliers, where  $\gamma \in R^{2N+1}$ . The saddle point of the Lagrangian function  $L(w, t, \xi_1, \xi_2)$  determines the solution to the constrained optimization problem.

Differentiating  $L(w, t, \xi_1, \xi_2)$  with respect to  $w$ ,  $t$ ,  $\xi_{1i}$ ,  $\xi_{2i}$  yields the following four equations:

$$\begin{aligned} \frac{\partial L(w, t, \xi_{1i}, \xi_{2i})}{\partial w} &= w - \gamma - \sum_{i=1}^L \alpha_i (2y_{1i} - 1) \vec{x}_{1i} \\ &\quad - \sum_{i=1}^L \beta_i (2y_{2i} - 1) \vec{x}_{2i}, \\ \frac{\partial L(w, t, \xi_{1i}, \xi_{2i})}{\partial t} &= \sum_{i=1}^L \alpha_i (2y_{1i} - 1) + \sum_{i=1}^L \beta_i (2y_{2i} - 1) - \eta, \\ \frac{\partial L(w, t, \xi_{1i}, \xi_{2i})}{\partial \xi_{1i}} &= C - \alpha_i - \mu_{1i}, \\ \frac{\partial L(w, t, \xi_{1i}, \xi_{2i})}{\partial \xi_{2i}} &= C - \beta_i - \mu_{2i}. \end{aligned} \quad (37)$$



The optimal value is obtained by setting the results of differentiating  $L(w, t, \xi_1, \xi_2)$  with respect to  $w$ ,  $t$ ,  $\xi_{1i}$ ,  $\xi_{2i}$  equal to zero. Thus,

$$w = \gamma + \sum_{i=1}^L \alpha_i (2y_{1i} - 1) \vec{x}_{1i} + \sum_{i=1}^L \beta_i (2y_{2i} - 1) \vec{x}_{2i}, \quad (38)$$

$$0 = \sum_{i=1}^L \alpha_i (2y_{1i} - 1) + \sum_{i=1}^L \beta_i (2y_{2i} - 1) - \eta, \quad (39)$$

$$C = \alpha_i + \mu_{1i}, \quad (40)$$

$$C = \beta_i + \mu_{2i}. \quad (41)$$

At each saddle point, for each Lagrange multiplier, the product of that multiplier with its corresponding constraint vanishes, as shown by

$$\alpha_i [(2y_{1i} - 1)(w^T \vec{x}_{1i} - t) + \xi_{1i} - 1] = 0, \quad \text{for } i = 1, 2, \dots, L, \quad (42)$$

$$\beta_i [(2y_{2i} - 1)(w^T \vec{x}_{2i} - t) + \xi_{2i} - 1] = 0, \quad \text{for } i = 1, 2, \dots, L, \quad (43)$$

$$\mu_{1i} \xi_{1i} = 0, \quad \text{for } i = 1, 2, \dots, L, \quad (44)$$

$$\mu_{2i} \xi_{2i} = 0, \quad \text{for } i = 1, 2, \dots, L. \quad (45)$$

By combining (40), (41), (44), and (45), (46) is gotten:

$$\begin{aligned} \xi_{1i} &= 0 & \text{if } \alpha_i < C, \\ \xi_{2i} &= 0 & \text{if } \beta_i < C. \end{aligned} \quad (46)$$

The corresponding dual problem is generated by introducing (38)–(41) into (36). Accordingly, the dual problem is formulated as follows.

Given the training samples  $\{(\vec{X}_i, m_i)\}_{i=1}^L$ , find the Lagrange multipliers  $\{\alpha_i\}_{i=1}^L$  that maximize the objective function

$$\begin{aligned} Q(\alpha, \beta) &= \sum_{i=1}^L (\alpha_i + \beta_i) - \frac{1}{2} \gamma^T \gamma - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j \\ &\quad \times (2y_{1i} - 1)(2y_{1j} - 1) \vec{x}_{1i}^T \vec{x}_{1j} \\ &\quad - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \beta_i \beta_j (2y_{2i} - 1)(2y_{2j} - 1) \vec{x}_{2i}^T \vec{x}_{2j} \\ &\quad - \gamma \sum_{i=1}^L \alpha_i (2y_{1i} - 1) \vec{x}_{1i} \\ &\quad - \gamma^T \sum_{i=1}^L \beta_i (2y_{2i} - 1) \vec{x}_{2i} - \sum_{i=1}^L \sum_{j=1}^L \alpha_i \beta_j \\ &\quad \times (2y_{1i} - 1)(2y_{2j} - 1) \vec{x}_{1i}^T \vec{x}_{2j} \end{aligned} \quad (47)$$

subject to the constraints

$$\begin{aligned} \sum_{i=1}^L \alpha_i (2y_{1i} - 1) + \sum_{i=1}^L \beta_i (2y_{2i} - 1) - \eta &= 0, \\ 0 \leq \alpha_i \leq C & \quad \text{for } i = 1, 2, \dots, L, \\ 0 \leq \beta_i \leq C & \quad \text{for } i = 1, 2, \dots, L, \\ \eta \geq 0, & \quad \gamma \geq 0, \end{aligned} \quad (48)$$

where  $C$  is a user-specified positive parameter and  $\vec{x}_{1i} = [X_{i-N}^{m_i}, X_{i-N+1}^{m_i}, \dots, X_i^{m_i}, \dots, X_{i+N}^{m_i}]^T$  and  $\vec{x}_{2i} = [X_{i-N}^{m_i+1}, X_{i-N+1}^{m_i+1}, \dots, X_i^{m_i+1}, \dots, X_{i+N}^{m_i+1}]^T$ .

#### 4.2. Nonlinear support vector machines for dichotomous WOS filters

When the number of training samples is large enough, (32) can be replaced as  $w^T \vec{x}_{1i} \geq 0$  because (1)  $\vec{x}_{1i}$  is a binary vector and (2) all possible cases of  $\vec{x}_{1i}$  are included by training samples. Then the problem is reformulated as follows.

Given the training samples  $\{(\vec{X}_i, m_i)\}_{i=1}^L$ , find an optimal value of the weight vector  $w$  and threshold  $t$  such that they satisfy the constraints

$$\begin{aligned} (2y_{1i} - 1)(w^T \vec{x}_{1i} - t) + \xi_{1i} &\geq 1, & \text{for } i = 1, 2, \dots, L, \\ (2y_{2i} - 1)(w^T \vec{x}_{2i} - t) + \xi_{2i} &\geq 1, & \text{for } i = 1, 2, \dots, L, \\ w^T \vec{x}_{1i} &\geq 0, & t \geq 0, \\ \xi_{1i}, \xi_{2i} &\geq 0, & \text{for } i = 1, 2, \dots, L, \end{aligned} \quad (49)$$

and such that the weight vector  $w$  and the slack variables  $\xi_{1i}$ ,  $\xi_{2i}$  can minimize the cost function:

$$\Phi(w, \xi_1, \xi_2) = \frac{1}{2} w^T w + C \sum_{i=1}^L (\xi_{1i} + \xi_{2i}). \quad (50)$$

Using the method of Lagrange multipliers and proceeding in a manner similar to that described in Section 4.1, the solution is gotten as follows:

$$\begin{aligned} w &= \sum_{i=1}^L \gamma_i \vec{x}_{1i} + \sum_{i=1}^L \alpha_i (2y_{1i} - 1) \vec{x}_{1i} + \sum_{i=1}^L \beta_i (2y_{2i} - 1) \vec{x}_{2i}, \\ 0 &= \sum_{i=1}^L \alpha_i (2y_{1i} - 1) + \sum_{i=1}^L \beta_i (2y_{2i} - 1) - \eta, \\ C &= \alpha_i + \mu_{1i}, \\ C &= \beta_i + \mu_{2i}. \end{aligned} \quad (51)$$

Then the dual problem is generated by introducing (51),

$$\begin{aligned} Q(\alpha, \beta, \gamma) &= \sum_{i=1}^L (\alpha_i + \beta_i) - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j (2y_{1i} - 1) \\ &\quad \times (2y_{1j} - 1) \vec{x}_{1i}^T \vec{x}_{1j} - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \gamma_i \gamma_j \vec{x}_{1i}^T \vec{x}_{2j} \\ &\quad - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \beta_i \beta_j (2y_{2i} - 1)(2y_{2j} - 1) \vec{x}_{2i}^T \vec{x}_{2j} \\ &\quad - \sum_{i=1}^L \sum_{j=1}^L \gamma_i \alpha_j (2y_{1j} - 1) \vec{x}_{1i}^T \vec{x}_{1j} \\ &\quad - \sum_{i=1}^L \sum_{j=1}^L \gamma_i \beta_j (2y_{2j} - 1) \vec{x}_{1i}^T \vec{x}_{2j} - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \beta_j \\ &\quad \times (2y_{1i} - 1)(2y_{2j} - 1) \vec{x}_{1i}^T \vec{x}_{2j}. \end{aligned} \quad (52)$$

The input data are mapped into a high-dimensional feature space by some nonlinear mapping chosen a priori. Let  $\varphi$  denote a set of nonlinear transformations from the input space  $R^m$  to a higher-dimensional feature space. Then (47) becomes

$$\begin{aligned}
Q(\alpha, \beta, \gamma) = & \sum_{i=1}^L (\alpha_i + \beta_i) - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j (2y_{1i} - 1) \\
& \times (2y_{1j} - 1) \varphi^T(\vec{x}_{1i}) \varphi(\vec{x}_{1j}) \\
& - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \gamma_i \gamma_j \varphi^T(\vec{x}_{1i}) \varphi(\vec{x}_{2j}) \\
& - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \beta_i \beta_j (2y_{2i} - 1) \\
& \times (2y_{2j} - 1) \varphi^T(\vec{x}_{2i}) \varphi(\vec{x}_{2j}) \\
& - \sum_{i=1}^L \sum_{j=1}^L \gamma_i \alpha_j (2y_{1j} - 1) \varphi^T(\vec{x}_{1i}) \varphi(\vec{x}_{1j}) \\
& - \sum_{i=1}^L \sum_{j=1}^L \gamma_i \beta_j (2y_{2j} - 1) \varphi^T(\vec{x}_{1i}) \varphi(\vec{x}_{2j}) \\
& - \sum_{i=1}^L \sum_{j=1}^L \alpha_i \beta_j (2y_{1i} - 1) (2y_{2j} - 1) \varphi^T(\vec{x}_{1i}) \varphi(\vec{x}_{2j}).
\end{aligned} \tag{53}$$

The inner product of the two vectors induced in the feature space can be replaced by the inner-product kernel denoted by  $K(x, x_i)$  and defined by

$$K(x, x_i) = \varphi(x) \cdot \varphi(x_i). \tag{54}$$

Once a kernel  $K(x, x_i)$  which satisfies Mercer's condition has been selected, the nonlinear model is stated as follows.

Given the training samples  $\{(\vec{X}_i, m_i)\}_{i=1}^L$ , find the Lagrange multipliers  $\{\alpha_i\}_{i=1}^L$  that maximize the objective function

$$\begin{aligned}
Q(\alpha, \beta, \gamma) = & \sum_{i=1}^L (\alpha_i + \beta_i) - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \alpha_i \alpha_j (2y_{1i} - 1) \\
& \times (2y_{1j} - 1) K(\vec{x}_{1i}, \vec{x}_{1j}) \\
& - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \gamma_i \gamma_j K(\vec{x}_{1i}, \vec{x}_{2j}) - \frac{1}{2} \sum_{i=1}^L \sum_{j=1}^L \beta_i \beta_j \\
& \times (2y_{2i} - 1) (2y_{2j} - 1) K(\vec{x}_{2i}, \vec{x}_{2j}) \\
& - \sum_{i=1}^L \sum_{j=1}^L \gamma_i \alpha_j (2y_{1j} - 1) K(\vec{x}_{1i}, \vec{x}_{1j}) - \sum_{i=1}^L \sum_{j=1}^L \gamma_i \beta_j \\
& \times (2y_{2j} - 1) K(\vec{x}_{1i}, \vec{x}_{2j}) \\
& - \sum_{i=1}^L \sum_{j=1}^L \alpha_i \beta_j (2y_{1i} - 1) (2y_{2j} - 1) K(\vec{x}_{1i}, \vec{x}_{2j})
\end{aligned} \tag{55}$$

subject to the constraints

$$\begin{aligned}
& \sum_{i=1}^L \alpha_i (2y_{1i} - 1) + \sum_{i=1}^L \beta_i (2y_{2i} - 1) - \eta = 0, \\
& 0 \leq \alpha_i \leq C \quad \text{for } i = 1, 2, \dots, L, \\
& 0 \leq \beta_i \leq C \quad \text{for } i = 1, 2, \dots, L, \\
& 0 \leq \gamma_i \quad \text{for } i = 1, 2, \dots, L,
\end{aligned} \tag{56}$$

where  $C$  is a user-specified positive parameter and  $\vec{x}_{1i} = [X_{i-N}^{m_i}, X_{i-N+1}^{m_i}, \dots, X_i^{m_i}, \dots, X_{i+N}^{m_i}]^T$  and  $\vec{x}_{2i} = [X_{i-N}^{m_i+1}, X_{i-N+1}^{m_i+1}, \dots, X_i^{m_i+1}, \dots, X_{i+N}^{m_i+1}]^T$ .

## 5. EXPERIMENTAL RESULTS

The ‘‘Lenna’’ and ‘‘Boat’’ images were used as training samples for a simulation. Dichotomous WOS filters were compared with adaptive neural filters, rank-order filter, and  $L_p$  norm WOS filter for the restoration of noisy images [20, 30, 31].

In the simulation, the proposed dichotomous WOS filters were used to restore images corrupted by impulse noise. The training results were used to filter the noisy images. With image restoration, the object function was modified in order to get an optimal solution. The learning steps are illustrated as follows.

*Step 1.* In  $i$ th training step, choose the input signal  $\vec{X}_i$  from a corrupted image and compare signal  $D_i$  from an uncorrupted image, where  $D_i \in \{0, 1, \dots, K - 1\}$ . The desired output  $Y_i$  is selected from input signal vector  $\vec{X}_i$  and  $Y_i = \{X_j \mid |X_j - D_i| \leq |X_k - D_i|, X_j, X_k \in \vec{X}_i\}$ .

*Step 2.* The training patterns  $\vec{x}_{1i}$  and  $\vec{x}_{2i}$  are gotten from input signal vector  $\vec{X}_i$  by using desired output  $Y_i$ .

*Step 3.* Calculating the distances  $S_{pi}$  and  $S_{qi}$ , where  $S_{pi}$  and  $S_{qi}$  are the distances between  $X_p, Y_i$  and  $X_q, Y_i$ . Note that  $X_p = \{X_j \mid Y_j - X_j \leq Y_i - X_k, X_j, X_k \in \vec{X}_i, \text{ and } X_j, X_k < Y_i\}$  and  $X_q = \{X_j \mid X_j - Y_j \leq X_k - Y_i, X_j, X_k \in \vec{X}_i, \text{ and } X_j, X_k > Y_i\}$ .

*Step 4.* The object function is modified by replacing  $\xi_{1i}$  and  $\xi_{2i}$  with  $S_{pi}\xi_{1i}$  and  $S_{qi}\xi_{2i}$ , where  $S_{pi}$  and  $S_{qi}$  are taken as the weight of the error.

*Step 5.* Applying the model of SVMs which is stated in Section 4 to get optimal solution.

A large dataset is generated when training data are obtained from a  $256 \times 256$  image. Nonlinear SVMs create unwieldy storage problems. There are various ways to overcome this including sequential minimal optimization (SMO), projected conjugate gradient chunking (PCGC), reduced support vector machines (RSVMs), and so forth [32–34]. In this paper, SMO was adopted because it has demonstrated outstanding performance.

Consider an example to illustrate how to generate the training data from the input signal. Let the input signal inside



FIGURE 2: (a) Original “Lenna” image; (b) “Lenna” image corrupted by 5% impulse noise; (c) “Lenna” image corrupted by 10% impulse noise; (d) “Lenna” image corrupted by 15% impulse noise.

the window of width 5 be  $\vec{X}_i = [240, 200, 90, 210, 180]^T$ . Suppose that the compared signal  $D_i$  which is selected from uncorrupted image is 208. The desired output  $Y_i$  is selected from input signal  $\vec{X}_i$ . According to the principle of WOS filters, the desired output is 210. Then,

$$\begin{aligned}\vec{x}_{1i} &= [T_{210}(240), T_{210}(200), T_{210}(90), T_{210}(210), T_{210}(180)]^T \\ &= [1, 0, 0, 1, 0]^T, \\ \vec{x}_{2i} &= [T_{211}(240), T_{211}(200), T_{211}(90), T_{211}(210), T_{211}(180)]^T \\ &= [1, 0, 0, 0, 0]^T,\end{aligned}\quad (57)$$

and  $y_{1i} = 1, y_{2i} = 0$ . The balance of training data is generated in the same way.

This section compares the dichotomous WOS filters with the adaptive neural filters in terms of three properties: time complexity, MSE, and convergence speed. Figures 2 and 3 present the training pairs, and Figures 4 and 6 present the images restored by the dichotomous WOS filters. Figures 5 and 7 show the images restored by the adaptive neural filters. Using SVMs on the dichotomous WOS filters with  $3 \times 3$  window width, the best near-optimal weight values for the test

images, which are corrupted by 5% impulse noise, are listed as follows:

$$\begin{aligned}\text{“Lenna”} &\Rightarrow \begin{pmatrix} 0.1968 & 0.2585 & 0.1646 \\ 0.1436 & 0.5066 & 0.1322 \\ 0.2069 & 0.2586 & 0.1453 \end{pmatrix} \\ \text{“Boat”} &\Rightarrow \begin{pmatrix} 0.1611 & 0.2937 & 0.1344 \\ 0.0910 & 0.5280 & 0.2838 \\ 0.1988 & 0.1887 & 0.1255 \end{pmatrix}.\end{aligned}\quad (58)$$

Notably, the weight matrix was translated row-wise in the simulation, that is,  $w_1 = w_{11}, w_2 = w_{12}, w_3 = w_{13}, w_4 = w_{21}, w_5 = w_{22}, w_6 = w_{23}, w_7 = w_{31}, w_8 = w_{32}, w_9 = w_{33}$ .

Three different kernel functions adopted in our experiments are polynomial function:  $(\text{gamma} * u^i * v + \text{coef})^{\text{degree}}$ , radial basis function:  $\exp(-\text{gamma} * \|u - v\|^2)$ , and sigmoid function:  $\tanh(\text{gamma} * u^i * v + \text{coef})$ , respectively. In our experiments, each element on training pattern is either 1 or 0. Suppose that three training patterns are  $\vec{x}_{k1} = [0, 0, 0, 0, 0, 0, 0, 0, 0]$ ,  $\vec{x}_{k2} = [0, 1, 0, 0, 0, 0, 0, 0, 0]$ , and  $\vec{x}_{k3} = [0, 0, 0, 1, 0, 0, 0, 0, 0]$ . Obviously, the difference between  $\vec{x}_{k1}, \vec{x}_{k2}$  and  $\vec{x}_{k1}, \vec{x}_{k3}$  cannot be distinguished when polynomial function or sigmoid function is adopted as



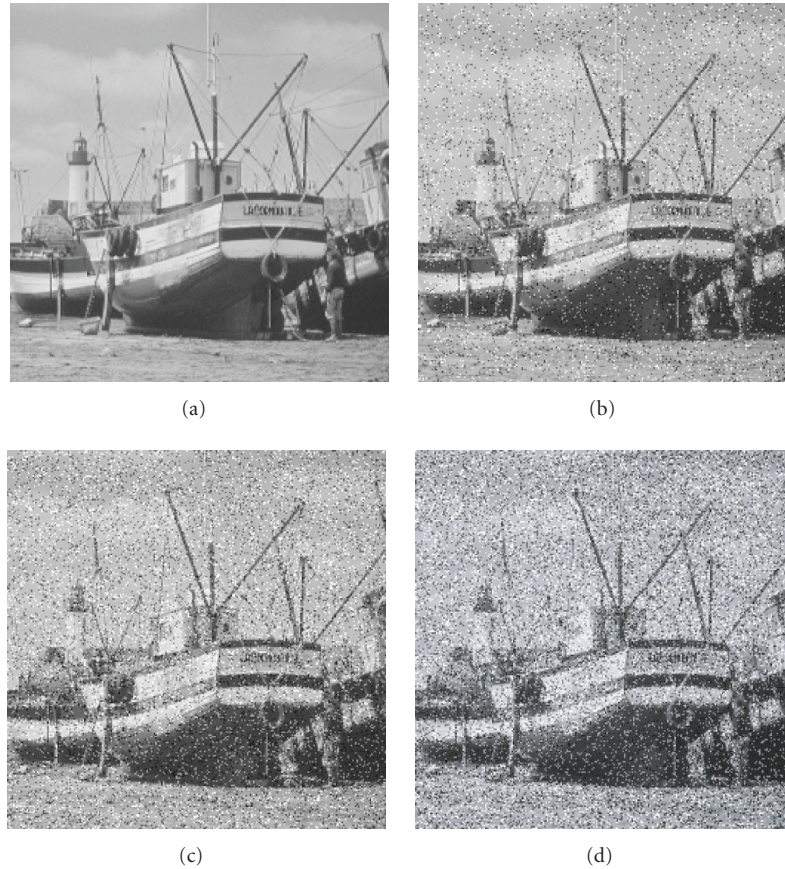


FIGURE 3: (a) Original “Boat” image; (b) “Boat” image corrupted by 5% impulse noise; (c) “Boat” image corrupted by 10% impulse noise; (d) “Boat” image corrupted by 15% impulse noise.



FIGURE 4: Using  $3 \times 3$  dichotomous WOS filter to restore (a) 5% impulse noise image; (b) 10% impulse noise image; (c) 15% impulse noise image.

kernel function. So in our experiments, only the radial basis function is considered. Besides, after testing with different values of gamma, 1 is adopted as the value of gamma in this experiment. Better classified ability and filtering performance are provided when the value of gamma is bigger than 0.5.

#### Time

If the computational time was  $T$  (time units) on each level, then the dichotomous WOS filters took only  $2T$  (time units) to filter 256 gray levels of data. However, the adaptive neural filters took  $255T$  (time units).

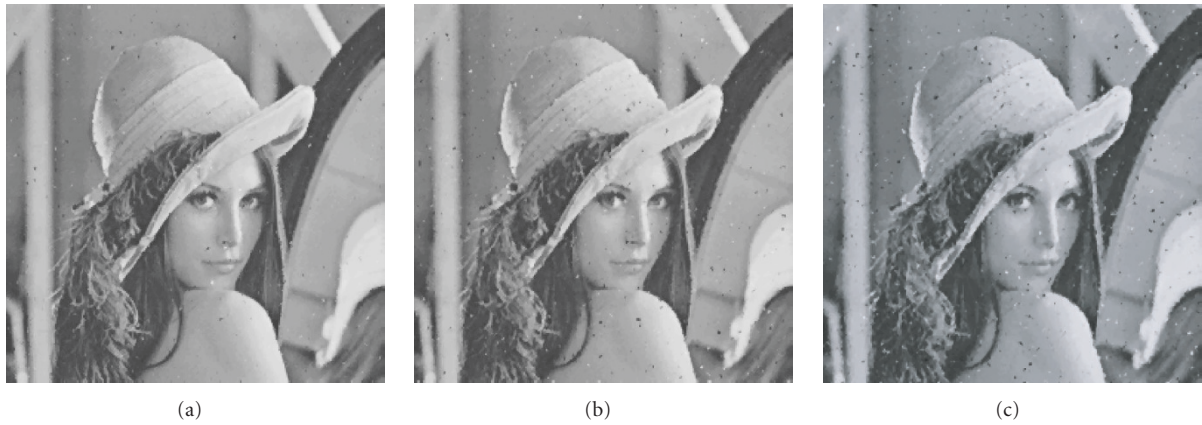


FIGURE 5: Using  $3 \times 3$  adaptive neural filter to restore (a) 5% impulse noise image; (b) 10% impulse noise image; (c) 15% impulse noise image.

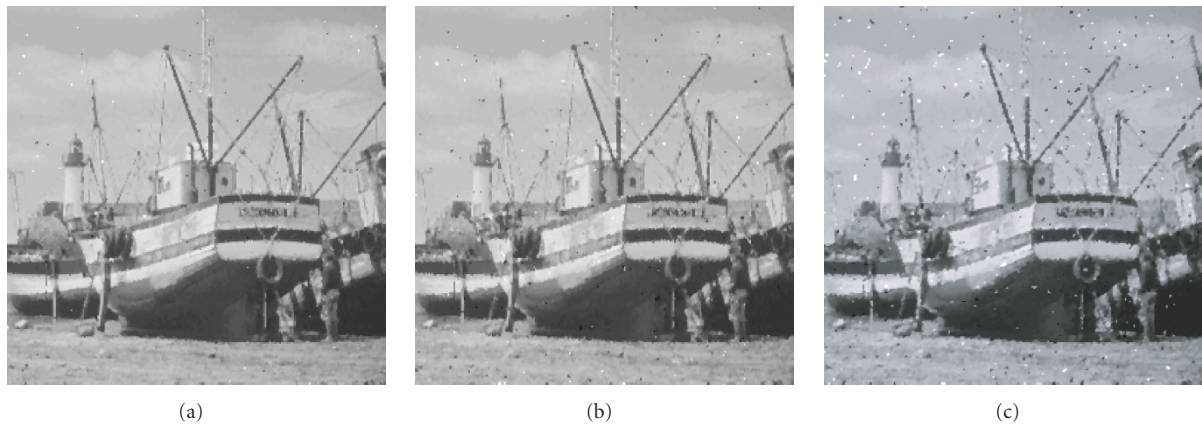


FIGURE 6: Using  $3 \times 3$  dichotomous WOS filter to restore (a) 5% impulse noise image; (b) 10% impulse noise image; (c) 15% impulse noise image.

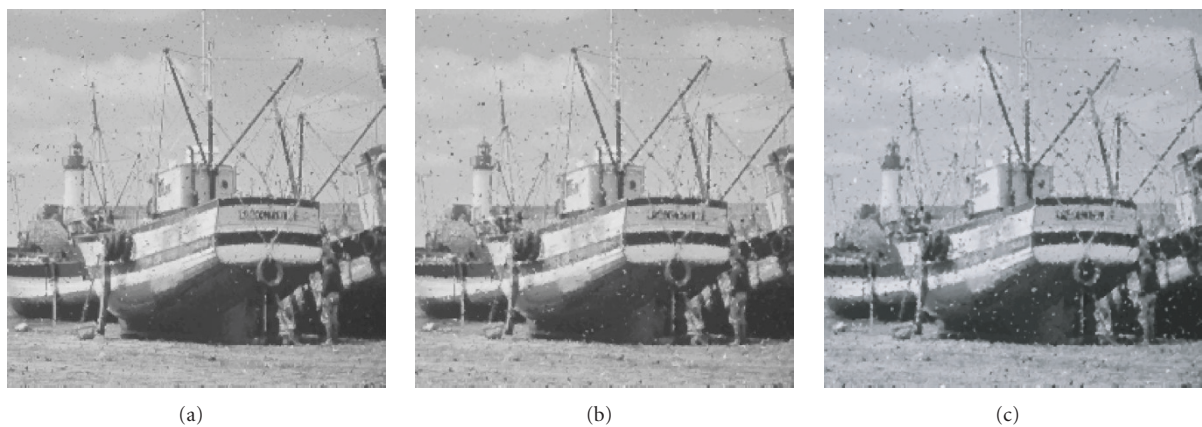


FIGURE 7: Using  $3 \times 3$  adaptive neural filter to restore (a) 5% impulse noise image; (b) 10% impulse noise image; (c) 15% impulse noise image.



TABLE 1: The comparisons of different filters' performance on impulsive noise image.

MSE errors	Measured error			
	WOS filter by SVMs	Adaptive neural filter	Rank-order filter	$L_p$ norm WOS filter
"Lenna" 5% noise	45	45	67.1	50.8
"Lenna" 10% noise	80.2	80	90.7	82.8
"Lenna" 15% noise	120.8	119	139.9	125.6
"Boat" 5% noise	95	95.6	155.7	105.1
"Boat" 10% noise	150.2	149	192.8	160.5
"Boat" 15% noise	208.8	206	256.9	218.4

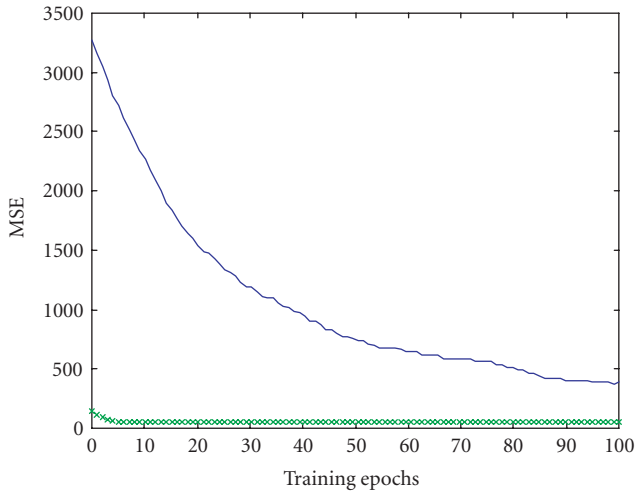


FIGURE 8: Converging speed of dichotomy WOS filter and adaptive neural filter: “-” indicates adaptive neural filter; “x” indicates dichotomous WOS filter.

## MSE

Table 1 lists the MSE values of the images restored with different filters. In this experiment, the adaptive neural filters used 256 levels to filter the data. In the simulation, nine-fold cross-validation was performed on the dataset to evaluate how well the algorithm generalizes to future data [35]. The ninefold cross-validation method extracts a certain proportion, typically 11%, of the training set as the tuning set, which is a surrogate of the testing set. For each training, the proposed method was applied to the rest of the training data to obtain a filter and the tuning set correctness of this filter was computed. Table 1 indicates that the dichotomous WOS filters performed as well as the adaptive neural filters. Both outperformed the rank-order filters and the  $L_p$  norm WOS filter.

Figure 8 compares convergence speeds. In Figure 8, the vertical axis represents MSE, while the horizontal axis represents the number of training epochs. Each unit of the horizontal axis represents 10 training epochs. Figure 8 reveals

that the dichotomous WOS filter converged steadily and more quickly than the adaptive neural filter.

In summary, the above comparisons revealed that dichotomous WOS filters outperformed adaptive neural filters, rank-order filters, and  $L_p$  norm WOS filter.

## 6. CONCLUSION

Support vector machines (SVMs), a classification algorithm for the machine learning community, have been shown to provide excellent performance on many applications. In this paper, SVMs are introduced into the design of WOS filters in order to improve performance.

WOS filters are special subset of stack filters. Each stack filter is based on a positive Boolean function and needs much computation time to achieve its Boolean computing. This makes the stack filter uneasy to use on application. Until now, the computation time has been only marginally improved by using the conventional design approach of stack filter or neural network. Although the adaptive neural filter can effectively remove noise of various kinds, including Gaussian noise and impulsive noise, its learning process involves a great deal of computational time. This work has proposed a new designing technique to approximate optimal WOS filters. The proposed technique, based on threshold composition, uses a dichotomous approach to reduce the Boolean computing from 255 levels to two levels. Then the technique of SVMs is used to get an optimal hyperplane to separate those two levels. The advantage of SVMs is that the risk of misclassifying is minimized not only with the examples in the training set, but also with the unseen examples of the test set. Our experimental results have showed that images were processed more efficiently than with an adaptive neural filter.

The proposed algorithm is designed to handle impulse noise and provided excellent performance on the images which contain impulse noise. We have experimented with the images which contain Gaussian noise, but the experimental results are unsatisfied. This reveals that a universal adaptive filter which can deal with any kind of noises simultaneously does not yet exist in the field of rank-ordered filters. This experimental result is consistent with the conclusion proposed by [36].

## ACKNOWLEDGMENT

This work is supported by National Science Council of Taiwan under Grant NSC93-2213-E-194-020.

## REFERENCES

- [1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.
- [2] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.
- [3] Y.-J. Lee and O. L. Mangasarian, "SSVM: a smooth support vector machine for classification," *Computational Optimization and Applications*, vol. 20, no. 1, pp. 5–22, 2001.
- [4] O. L. Mangasarian, "Generalized support vector machines," in *Advances in Large Margin Classifiers*, A. J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, Eds., pp. 135–146, MIT Press, Cambridge, Mass, USA, 2000.
- [5] O. L. Mangasarian and D. R. Musicant, "Successive overrelaxation for support vector machines," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1032–1037, 1999.
- [6] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.
- [7] G. Guo, S. Z. Li, and K. L. Chan, "Support vector machines for face recognition," *Image and Vision Computing*, vol. 19, no. 9–10, pp. 631–638, 2001.
- [8] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.
- [9] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.
- [10] R. Yang, M. Gabbouj, and P.-T. Yu, "Parametric analysis of weighted order statistics filters," *IEEE Signal Processing Letters*, vol. 1, no. 6, pp. 95–98, 1994.
- [11] P.-T. Yu, "Some representation properties of stack filters," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2261–2266, 1992.
- [12] P.-T. Yu and R.-C. Chen, "Fuzzy stack filters—their definitions, fundamental properties, and application in image processing," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 838–854, 1996.
- [13] P.-T. Yu and E. J. Coyle, "The classification and associative memory capability of stack filters," *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2483–2497, 1992.
- [14] P.-T. Yu and E. J. Coyle, "Convergence behavior and N-roots of stack filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 9, pp. 1529–1544, 1990.
- [15] P.-T. Yu and W.-H. Liao, "Weighted order statistics filters—their classification, some properties, and conversion algorithm," *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2678–2691, 1994.
- [16] C. Chakrabarti and L. E. Lucke, "VLSI architectures for weighted order statistic (WOS) filters," *Signal Processing*, vol. 80, no. 8, pp. 1419–1433, 2000.
- [17] S. W. Perry and L. Guan, "Weight assignment for adaptive image restoration by neural networks," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 156–170, 2000.
- [18] H.-S. Wong and L. Guan, "A neural learning approach for adaptive image restoration using a fuzzy model-based network architecture," *IEEE Transactions on Neural Networks*, vol. 12, no. 3, pp. 516–531, 2001.
- [19] L. Yin, J. Astola, and Y. Neuvo, "Optimal weighted order statistic filters under the mean absolute error criterion," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '91)*, vol. 4, pp. 2529–2532, Toronto, Ontario, Canada, April 1991.
- [20] L. Yin, J. Astola, and Y. Neuvo, "A new class of nonlinear filters—neural filters," *IEEE Transactions on Signal Processing*, vol. 41, no. 3, pp. 1201–1222, 1993.
- [21] L. Yin, J. Astola, and Y. Neuvo, "Adaptive multistage weighted order statistic filters based on the backpropagation algorithm," *IEEE Transactions on Signal Processing*, vol. 42, no. 2, pp. 419–422, 1994.
- [22] P. D. Wendt, E. J. Coyle, and N. C. Gallagher, "Stack filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 898–911, 1986.
- [23] M. J. Avedillo, J. M. Quintana, and E. Rodriguez-Villegas, "Simple parallel weighted order statistic filter implementations," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '02)*, vol. 4, pp. 607–610, May 2002.
- [24] A. Gasteratos and I. Andreadis, "A new algorithm for weighted order statistics operations," *IEEE Signal Processing Letters*, vol. 6, no. 4, pp. 84–86, 1999.
- [25] H. Huttunen and P. Koivisto, "Training based optimization of weighted order statistic filters under breakdown criteria," in *Proceedings of the International Conference on Image Processing (ICIP '99)*, vol. 4, pp. 172–176, Kobe, Japan, October 1999.
- [26] P. Koivisto and H. Huttunen, "Design of weighted order statistic filters by training-based optimization," in *Proceedings of the 6th International Symposium on Signal Processing and Its Applications (ISSPA '01)*, vol. 1, pp. 40–43, Kuala Lumpur, Malaysia, August 2001.
- [27] S. Marshall, "New direct design method for weighted order statistic filters," *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 151, no. 1, pp. 1–8, 2001.
- [28] O. Yli-Harja, J. Astola, and Y. Neuvo, "Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 395–410, 1991.
- [29] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, Mass, USA, 1999.
- [30] J. Poikonen and A. Paasio, "A ranked order filter implementation for parallel analog processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 5, pp. 974–987, 2004.
- [31] C. E. Savin, M. O. Ahmad, and M. N. S. Swamy, " $L_p$  norm design of stack filters," *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1730–1743, 1999.
- [32] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, Mass, USA, 2002.
- [33] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, UK, 1981.
- [34] Y.-J. Lee and O. L. Mangasarian, "RSVM: Reduced Support Vector Machines," in *Proceedings of the 1st SIAM International Conference on Data Mining*, Chicago, Ill, USA, April 2001.
- [35] M. Stone, "Cross-validated choice and assessment of statistical predictions," *Journal of the Royal Statistical Society*, vol. B36, pp. 111–147, 1974.
- [36] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: a tutorial," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 3, pp. 157–192, 1996.

**Chih-Chia Yao** received his B.S. degree in computer science and information engineering from National Chiao Tung University, in 1992, and M.S. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 1994. He is a Lecturer in the Department of Information Management, Nankai College, Nantou, Taiwan. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering, National Chung Cheng University. His research interests include possibility reasoning, machine learning, data mining, and fuzzy inference system.



**Pao-Ta Yu** received the B.S. degree in mathematics from National Taiwan Normal University in 1979, the M.S. degree in computer science from National Taiwan University, Taipei, Taiwan, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, Ind, USA, 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 e-learning, neural networks and fuzzy systems, nonlinear filter design, intelligent networks, and XML technology.

