Abstract—A partition-based adaptive vector filter is proposed for the restoration of corrupted digital color images. The novelty of the filter lies in its unique three-stage adaptive estimation. The local image structure is first estimated by a series of center-weighted reference filters. Then the distances between the observed central pixel and estimated references are utilized to classify the local inputs into one of preset structure partition cells. Finally, a weighted filtering operation, indexed by the partition cell, is applied to the estimated references in order to restore the central pixel value. The weighted filtering operation is optimized off-line for each partition cell to achieve the best tradeoff between noise suppression and structure preservation. Recursive filtering operation and recursive weight training are also investigated to further boost the restoration performance. The proposed filter has demonstrated satisfactory results in suppressing many distinct types of noise in natural color images. Noticeable performance gains are demonstrated over other prior-art methods in terms of standard objective measurements, the visual image quality and the computational complexity.

Index Terms—Center-weighted vector median (CWVM) filter, constrained least mean-square (LMS) algorithm, digital color image restoration, partition-based adaptive vector (PBVTVM) filter.

I. INTRODUCTION

ONLINEAR vector filtering techniques have generated much research interest in the last decade due to its importance in color image restoration. Numerous filtering techniques proposed to date are based on multivariate order statistics [1]–[4], which take the advantage of color inter-channel dependence and avoid unpleasant drawbacks of component-wise filtering techniques, i.e., pixel value rearranging and chromatic shifting [5], [6]. The most well-known vector filters include the vector median filters (VMF) [7], the vector directional filter (VDF) [8], and the directional-distance filter (DDF) [9]. They perform well in suppressing impulse noise and outliers, but introduce blurring artifacts in edge and detail areas which feature high spacial frequency contents and variations. Weighted nonlinear vector filtering techniques have been proposed to achieve better performance in noise suppression and detail preservation, which include distance-based weighted vector filters [10]–[13], weighted VDFs [14], [15], and the generalized selection weighted vector filters [16]. Adaptive vector filters have been advanced for their effectiveness with respect to different types of noise distributions and image structures, which include filters utilizing local order statistics information [17]–[22], based on gradient information [23], with peer group classification [23], and using a digital path approach [25], to name a few. In the case where noise model cannot be known a priori, fuzzy vector filtering techniques have been developed and achieved robust performance [26]–[31]. Recently, a class of chromatic filters worked in the YCbCr color space is also proposed to achieve better chromatic smoothness [32]. Please refer to [33] for a comprehensive review on color image restoration techniques.

The fact that different types of noise contaminate color images in distinct ways poses a major challenge for the vector filtering adaption [6], [33], [34]. Impulse noise destroys only a small portion of an image and leaves other pixels noise free, while additive noise contaminates every image pixel with a certain type of statistical distribution. Detection-based vector filtering techniques, such as the adaptive vector median filter (AVMF) [12], the adaptive vector LUM smoother (AVLUM) [13], the modified weighted vector median (MWVM) filter [11], and the selection center-weighted vector directional filter (SCWVDF)/adaptive center-weighted vector directional filter (ACWVDF) [14], were specially designed to remove impulse noise. They utilize a series of weighted vector median filters to perform binary noise detection, and switch the outputs between an identity filter and a weighted vector median filter according to the detection results. Such a structure, while efficient for impulse noise removal, is inappropriate in dealing with additive noise or mixed noise contamination. In comparison, many adaptive vector filters, such as the adaptive nearest-neighbor multichannel filter (ANNF) [17], the adaptive vector directional filter (AVDF) [18], and the multichannel II filter [23], yield output by a robust weighting estimation based on all image vectors in the filter window. They are designed to cope with additive and mixed noise corruption, but at the cost of possible image edges and details smearing. Some fuzzy vector filtering techniques [28], [31] have tried to combine different types of standard vector filters together by a large number of fuzzy rules. However, the systematic optimization of such an approach remains as an open problem. Others techniques
[26], [27], [29] proposed a group of fuzzy similarity functions within the weighted filter structure to enhance the robustness of their performance. However, the selection of optimal similarity functions is left to the user to decide, more often than not, in an ad hoc manner. It is still a challenging task to design an efficient vector filtering framework to cope with different types of noise in color images, i.e., impulse noise, additive noise, as well as mixed noise, with its performance comparable to that of the state-of-the-art filtering techniques designed for each of these distinct types of noise.

In this paper, a novel partition-based vector filtering framework is proposed for restoring color images contaminated by different types of noise. The novelty of the filter lies in its unique three-stage adaptive estimation, which consists of a novel center-weighted trimmed vector median (CWTVM) filter, a distance-based classification scheme, and a partition indexed weighted filtering operation. The CWTVM filter is used to provide robust reference estimates. The distance-based classification scheme is used to map the local inputs into one of preset structure cells. The weighted filtering operation is designed to provide the best reconstruction with regard to the specific structure partition classification. Off-line training is used to determine the filtering weights for each structure partition cell to achieve the best computation efficiency and performance, using a constrained least-mean-square (LMS) algorithm well-defined in the vector space. The proposed vector filtering structure, named hereafter as the partition-based trimmed vector median (PBTVM) filter, has achieved superior performance to a number of well-known benchmarks in removing different types of impulse noise, additive noise and mixed noise, in terms of standard objective measurements and the subjective quality.

The paper is structured as follows. The central weighted reference filter is first formulated in Section II. Then in Section III, the framework of the PBTVM filter is described in detail. Section IV addresses the structure partitioning, the weight training, and the recursive implementation of the filtering/training. A number of experimental results are presented in Section V and a brief conclusion is drawn in Section VI.

II. CENTER-WEIGHTED REFERENCE FILTER

A. Review of the CWVM Filter

A weighted vector median (WVM) filter [10], [35] has been widely used in color image restoration for its high flexibility in noise suppression and detail preservation, and a number of algorithms [16], [36], [37] have been developed to optimize its multiple filtering weights. Center-weighted vector median (CWVM) filter [11]–[15] is a special type of the WVM filter where only the weight of the CP is adjustable. It inherits the flexibility of the WVM filter but greatly reduces the complexity of implementation and optimization. In this section, the structure of the CWVM filter is reviewed along with associated terminologies that will be used in later sections.

Let \( C = \{ c_n = (c_{1n}, c_{2n}) \mid 1 \leq c_1 \leq H, 1 \leq c_2 \leq W \} \) be the pixel coordinates of an RGB color image, where \( H \) and \( W \) are the height and the width of the image, respectively. At each coordinate \( c_n \in C \), a multivariate \( x(c) = [x_{R(c)}, x_{G(c)}, x_{B(c)}]^T \) is used to represent its pixel value, and a square filter window, \( W(c) = \{ x_1(c), x_2(c), \ldots, x_N(c) \} \), is also defined, which centers at the coordinate \( c \) and contains \( N \) pixel samples (where \( N \) is an odd integer). The pixel samples of the window are indexed in a lexicographic scan order, so that the central pixel (CP) of the window, \( x(c) \), always equals to \( x_{2k+1}(c) \). Fig. 1 illustrates the indexed window pixels of a \( 3 \times 3 \) filter window, where \( x(c) = \hat{x}_0(c) \) is the CP of the window.

Based on the terminologies above, the output of a CWVM filter with a positive integer central weight, \( k \), is given by [10]–[13], [36]

\[
y_k(c) = x_m(c) \in W(c) \mid m = \arg \min_{1 \leq i \leq N} \{ R_i^k \}
\]

where \( R_i^k \) is the center-weighted aggregated vector distance (CW-AVD) given by

\[
R_i^k = (2k - 1) ||x_i(c) - x(c)|| + \sum_{j=1, j \neq i}^{N} ||x_j(c) - x_i(c)||
\]

and \( || \cdot || \) denotes the \( L_2 \) norm (Euclidian distance) in this paper, whereas other vector measures, such as the \( L_1 \) norm [7], the generated Minkowski metric [6], the vector directional distance [16], or other measures [33], [38], can be used here, as well.

The central weight of the CWVM filter can be adjusted efficiently to achieve different degrees of noise suppression and image preservation. When the central weight \( k = 1 \), the CWVM filter reduces to a standard VMF filter, which is powerful for noise suppression but smears small details in the filter window. As the value of \( k \) increases, the detail preservation of the filter is improved while the noise suppression performance is degraded. The CWVM filter becomes an identity filter when \( k \geq (N + 1)/2 \). Proofs of such properties of the CWVM filter can be found in [10], [14]. An example of the outputs of the CWVM filter is provided in Appendix.

B. Formulation of the CWTVM Filter

Despite the advantages of the CWVM filter and its success in color image restoration, it has been found in our experiments that a CWVM filter, if working with a relatively high central weight value, often turns out to be oversensitive to impulse noise, while its performance decreases as the value of \( k \) increases. A novel partition-based trimmed vector median (CWTVM) filter is proposed in this paper to improve the performance of the CWVM filter.
noise. This renders the CWVM filter an inappropriate reference filter for the robust local structure classification in Section III. Therefore, a novel CWTVM filter, which with high central weights can achieve more robust performance than the CWVM filter, is proposed in this subsection.

The structure of the CWTVM filter is distinct from that of the CWVM filter in three aspects.

1. Before the CW-AVD calculation, all pixels of a filter window are ranked in an ascending order according to their Euclidian distances to the CP of the window.

2. Then a trimming technique is applied to the pixels of the window according to the central weight value. Pixels which have been ranked beyond a certain CP-local distance order are discarded.

3. Finally, a modified CW-AVD function is applied to the remaining pixels of the window to find the filtering output, using a central weighting scheme different from that of the CWVM filter.

Denoting \( W(c) \) as a filter window centered at coordinate \( c \), the Euclidian distances between the CP and all \( N \) pixels of the window are given by

\[
d_j = ||x_j(c) - x(c)||, \quad 1 \leq j \leq N
\]  

(3)

where \( x(c) \) represents the CP of the window. \( x_j(c) \in W(c) \) denotes a pixel of the window, and \( x(c) \triangleq x_{x(c)} \) as in the previous subsection. The \( d_j \)’s are ranked in an ascending order such that \( d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(i)} \leq \cdots \leq d_{(N)} \) (Note that the coordinate variable \( c \) is omitted for \( d_{(j)} \) and \( d_{(i)} \) for simplicity). Accordingly, the pixels of the filter window are re-expressed as

\[
W'(c) = \{x_{(i)}(c) \mid i = 1, 2, \ldots, N\}
\]  

(4)

where \( x_{(i)}(c) \) denotes a pixel of the window, which is indexed by its CP-local distance ranking order, that is,

\[
x_j(c) \to x_{(i)}(c) \iff d_j \to d_{(i)},
\]  

(5)

where the symbol “\( \to \)” represents a mapping from an original value to its ranked version. Note that \( W'(c) \) and \( W(c) \) consist of the same set of pixels, but in different orders.

Based on the ranked pixel set \( W'(c) \), the output of the CWTVM filter with an integer central weight, \( k \), is defined as

\[
y_k(c) = x_{(m)}(c) \in W'(c) \mid m = \arg \min_{1 \leq i \leq N} \left\{ \hat{R}_k^i \right\}
\]  

(6)

where \( \hat{R}_k^i \) is the modified CW-AVD function (or MCW-AVD) given by

\[
\hat{R}_k^i = (k-1)||x_{(i)}(c) - x(c)|| + \sum_{j=1}^{N-(k-1)} ||x_{(j)}(c) - x_{(i)}(c)||.
\]  

(7)

The proposed CWTVM filter shares a number of properties with its counterpart, the CWVM filter. The CWTVM filter reduces to a standard VFM filter when \( k = 1 \), or an identity filter when \( k \geq \frac{N+1}{2} \). With \( k \) increased in the range of \([1, (N + 1)/2]\), the detail preservation of the CWTVM filter is improved whereas its noise suppression performance degraded. However, the CWTVM filter can achieve more robustness against impulse noise than what the CWVM filter does when the central weight increases, due to its novel CP-distance trimmed technique given in (7). Based on the random distribution of impulse noise, such a technique is able to trim away some noise even before the multivariate aggregation ordering is applied.

Evaluation of the CWTVM filter is conducted on a wide range of impulse corrupted natural color images. Table I lists the results of the CWTVM and CWVM filters on a 512 × 512-pixel 24-bit RGB image, Lena, corrupted by different ratios of random impulse noise, where a 3 × 3 filter window is used. Noticeable gains are observed when the CWTVM filter works with the central weight \( k = 3 \) or 4 and the noise ratio is relatively high. The mechanism leading to the performance improvement is analyzed in Example 2 of Appendix. Such an improvement will greatly benefit the local structure estimation and classification proposed in Section III. The performance of the CWTVM filter is slightly lower than that of the CWVM filter when the central weight \( k = 1 \) or 2. However, its influence on the proposed PBTVM filter is negligible as long as the CWTVM filter outputs reference estimates with reduced noise.

### III. PARTITION-BASED VECTOR FILTERING STRUCTURE

#### A. Review of Switch-Based Vector Median Filters

A number of switch-based vector median filters [11], [13], [14] have been reported in recent years, which take the advantages of the CWVM filter to detect impulse corruptions and to preserve fine image structures. Given a filter window \( W(c) \) centered at the pixel \( x(c) \) containing \( N \) pixel samples, these filters first utilize a series of CWVM filters with their central weights \( k = 1, 2, \ldots, (N - 1)/2 \), respectively, to obtain a set of reference estimates, \( \{y_k(c) \mid k = 1, 2, \ldots, (N - 1)/2\} \), from the pixels of the window, where \( y_k(c) \) denotes the output of a
CWVM filter with a central weight $k$. Then the distance between the CP and each of the reference estimates is measured as

$$c_k(c) = \|y_k(c) - x(c)\|, \quad k = 1, 2, \ldots, \frac{(N - 1)}{2}$$

(8)

where $\| \cdot \|$ represents the Euclidean distance for the AVLUM filter [13] and the MWVM filter [11], or the directional distance for the SCWVDF/ACWVDF filter [14]. Finally, a group of preset thresholds, $\{T_{0k} \mid k = 1, 2, \ldots, \frac{(N - 1)}{2}\}$, is used to evaluate the irregularity of the CP-reference distance $c_k(c)$, and accordingly, a binary decision mechanism is activated to determine the filter output, $\hat{x}(c)$, as follows [13], [14]

$$\hat{x}(c) = \begin{cases} x(c), & \text{if all } c_k(c) \leq T_{0k}, 1 \leq k \leq \frac{N-1}{2} \\ \gamma_m(c), & m = \min\{k \mid c_k(c) > T_{0k}\} \end{cases}$$

(9)

Despite their state-of-the-art performance in impulse suppression and structure preservation, these switch-based vector median filters still have a few drawbacks.

1) The accuracy of their binary noise detection heavily relies on the threshold values ($T_{0k}$s) and the condition (e.g., impulse noise model and image structure) of the threshold optimization. It is hard for these filters to cope with different impulse models or varying image characteristics.

2) Although their switch-based reconstruction mechanism works efficiently for the impulse noise removal, such a selective mechanism is inappropriate for suppressing other distinct types of noise, such as additive noise or mixed noise which contaminates original images.

B. Formulation of the PBTVM Filter

A new filtering framework is proposed in this section to eliminate the drawbacks associated with the existing switch-based vector median filters. The new filter, named hereafter as the partition-based trimmed vector median (PBTVM) filter, combines the robust CWTVM filter of Section II-B with the partition-based weighted filtering techniques developed in [40], [41], inheriting the merits from both techniques. A group of estimates are first produced by the CWTVM reference filter. Then, a structure classifier is formulated to map the inputs into one of the partition cells. Finally, a weighted filtering operation is activated to achieve the best noise suppression and image preservation for the classified structure. The new PBTVM filter involves neither binary noise detection nor selective filtering mechanism. The schematic diagram of the new filter is presented in Fig. 2. Details of the CP-reference based structure classification and the partition-based weighted filtering are described in the following sections.

1) CP-Reference-Based Structure Classification: Given a filter window, $\mathbf{W}(c)$, which centers at the CP, $\mathbf{x}(c)$, and contains $N$ pixel samples, a series of CWTVM filters with their central weights $k = 1, 2, \ldots, K$, respectively, are produced to produce a set of reference estimates, $\{\hat{y}_k(c) \mid k = 1, 2, \ldots, K\}$, where $\hat{y}_k(c)$ is the CWTVM filtering output defined by (6), and $K = \frac{(N - 1)}{2}$. After the reference filtering, the distance between each of the estimates and the CP of the window is measured as

$$\hat{c}_k(c) = \|\hat{y}_k(c) - x(c)\|, \quad k = 1, 2, \ldots, K$$

(10)

and a distance vector, $\hat{\mathbf{e}}(c)$, is formulated as follows:

$$\hat{\mathbf{e}}(c) = [\hat{e}_1(c), \hat{e}_2(c), \ldots, \hat{e}_K(c)]$$

(11)

The distance vector $\hat{\mathbf{e}}(c)$ as a whole contains more local structure information than any of its component, $\hat{c}_k(c)$. In order to reveal most of the structured information in the vector $\hat{\mathbf{e}}(c)$, a structure classifier, $\mathcal{C}(\cdot)$, is formulated as a function of the vector $\hat{\mathbf{e}}(c)$. The classifier maps the inputs from an $\mathbb{R}^K$ dimensional vector space to one of $M$ mutually exclusive partition cells as follows:

$$\Psi_i = \{\hat{\mathbf{e}}(c) \in \mathbb{R}^K : \mathcal{C}[\hat{\mathbf{e}}(c)] = i\}$$

(12)

where the partition cells, $\{\Psi_i \mid i = 1, 2, \ldots, M\}$, satisfy $\bigcup_i \Psi_i = \mathbb{R}^K$ and $\Psi_i \cap \Psi_j = \emptyset$ for $i \neq j$.

Various partition methods, such as the scalar quantization (SQ) [42], the fuzzy quantization (FQ) [43], or the vector quantization (VQ) [44] can be used to construct the classifier. In this paper, the scalar quantization is selected due to its computational efficiency and high robustness. Given the distance vector $\hat{\mathbf{e}}(c)$, a group of decision levels (thresholds), $\{T_{k,l} \mid l = 0, 1, \ldots, L\}$, are defined for each of its components, $\hat{e}_k(c) \mid 1 \leq k \leq K$. The levels are formulated in a monotonically ascending order so that $T_{k,l} \leq T_{k,l+1}$ for any $0 \leq l \leq L - 1$, where $L$ is a preset value that controls the number of decision levels for each $\hat{e}_k(c)$, and accordingly, the total number of partition cells, i.e., $M = L^K$.

Following these definitions, the classifier $\mathcal{C}(\cdot)$ can be expressed by a partition index, $p(c)$, which is given by

$$p(c) = 1 + \sum_{k=1}^{K} L^{(K-k)} \psi_k$$

(13)
where for $k = 1, 2, \ldots, K$

$$\psi_k = q_k \text{ if } \ell_k(c) \in [T_{k,q}, T_{k,q+1}), \quad 0 \leq q \leq L - 1. \quad (14)$$

An example is provided in Appendix to demonstrate the calculation of the partition index $p(c)$ for $K = 3$ and $L = 2$.

The proposed classification method is distinct from the noise detection methods used in the existing switch-based vector median filters [11], [13], [14]. In those filters, reference estimates are produced by the CWVM filters, and the CP-reference distances are compared with a set of thresholds individually to contribute a binary noise decision. However, in the proposed method, the new robust CWTVM filter is used for the reference estimate generation, and the classifier is formulated on a multivariate distance space which maps the local inputs to one of the partition cells. Moreover, each of these partition cells represents a specific local structure, which can be restored by an optimal weighted filtering operation, regardless of whether it is corrupted and how the corruption occurs.

2) Partition-Based Weighted Filtering: For the given filter window, $W(c)$, all the reference estimates from the CWTVM filter, $\hat{y}_k(c) \mid k = 1, 2, \ldots, K$, and the input CP, $x(c)$, are used to reconstruct the original CP value, $\hat{x}(c)$, via a weighted filtering operation given by

$$\hat{x}(c) = w_{i,0}x(c) + \sum_{k=1}^{K} w_{i,k}\hat{y}_k(c)$$

$$\equiv w^T_i \odot \hat{y}(c) \quad (15)$$

where “$\odot$" denotes an unconventional “dot” product between a row vector and a column vector. As shown in (15), each element of the row vector is a scaler while each element of the column vector is a $3 \times 1$ vector, and the number of elements (scalers) of the row vector must be the same as that of elements ($3 \times 1$-column-vectors) of the column vector for “$\odot$” to apply. Between the elements of the two vectors, the product is a product between a scaler and a vector (or matrix) which has been defined in the conventional matrix algebra. The row vector

$$w^T_i = [w_{i,0}, w_{i,1}, w_{i,2}, \ldots, w_{i,K}] \quad (16)$$

is the weight vector for the filtering operation, where the value of $i = p(c)$ indicates the index to the classified partition cell, and the column vector

$$\hat{y}(c) = [\hat{y}_i^T(c), \hat{y}_2^T(c), \ldots, \hat{y}_K^T(c)]^T \quad (17)$$

consists of all inputs of the weighted filtering operation. To facilitate further discussion, the weight vector is rewritten as $w_i = [w_{i,0}, \ldots, w_{i,K}]^T$, where $\tilde{w}_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,K}]^T$ is the reduced weight vector. Accordingly, the inputs of the weighted filtering operation is rewritten as $\hat{y}(c) = [\hat{y}_i^T(c), \hat{y}_2^T(c), \ldots, \hat{y}_K^T(c)]^T$, where $\hat{y}(c) = [\hat{y}_i^T(c), \hat{y}_2^T(c), \ldots, \hat{y}_K^T(c)]^T$ is the reduced filtering input vector.

Local invariance constraint [39] is applied to the weight vector to guarantee an unbiased reconstruction, which leads to

$$w_{i,0} = 1 - \tilde{w}_i^T1$$

where $1 = [1, 1, \ldots, 1]^T$ is a $K \times 1$ unitary vector. As such, the output of the PBCTVM filter defined by (15) is rewritten as

$$\hat{x}(c) = w_{i,0}x(c) + \tilde{w}_i^T \odot \hat{y}(c). \quad (19)$$

Based on its flexible weight configurations, the proposed filtering structure can be of any between a highly nonlinear median filter and a linear weighted averaging filter, which provides an efficient solution to suppress distinct types of noise contaminations. The structure includes the switch-based vector median filters [11], [13] as a special case, provided that the filtering weights, $w_{i,0}$, are switched between one and zero according to a binary noise detection. However, there are two aspects that render the new structure distinct from the existing adaptive vector filters [17], [26]. First, the inputs of the weighted filtering are the reference estimates from a series of robust CWTVM filters, while the inputs of the existing adaptive vector filters are pixel samples produced by a local multivariate ordering. Second, the proposed filtering weight vector is trained off-line and selected on-line according to the structure classification, while the weights of those adaptive filters are generated on-line by a robust statistics estimator or a fuzzy membership function preset for the entire image.

IV. PARTITION THRESHOLD FORMULATION AND FILTERING WEIGHT TRAINING

A. Partition Threshold Formulation

From a theoretic point of view, the partitioning thresholds used in (14) and filtering weights related to each partition cell should be jointly optimized to achieve the best restoration performance. However, such a nonlinear coupled optimization process often incurs a high computational cost, and easily leads to local minima [44], [45]. In this paper, a decoupled optimization technique is developed to avoid aforesaid issues, which is depicted as follows.

1) For initialization, the total number of partition cells and the starting threshold values are specified using a heuristic approach and prior knowledge.

2) Based on the initialized partition thresholds, the filtering weights are updated using the constrained LMS algorithm described in Section IV-B and a training image contaminated by a specified noise source.

3) Given the updated filtering weights, the Simplex minimization method [46] is used to optimize the threshold values associated with each $\ell_k(c)$ independently. The process terminates when all the threshold updating sizes are smaller than 0.5; else

4) Repeat from Step 2) using the updated thresholds.

It is clear that the threshold set obtained by the proposed optimization process theoretically may lead to a suboptimal solution. However, by tuning threshold values on a range of test im-
TABLE II
PROPOSED THRESHOLD, \(T_{\lambda,2}\), FOR THE PARTITIONING OF THE
DISTANCE VECTOR SPACE OF A 3 × 3 FILTER WINDOW,
WHERE 1 ≤ \(k\) ≤ 4 AND 0 ≤ \(l\) ≤ 8

<table>
<thead>
<tr>
<th>(k)</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>28.1</td>
<td>46.3</td>
<td>59.1</td>
<td>81.6</td>
<td>98.2</td>
<td>106.8</td>
<td>127.8</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>9.3</td>
<td>24.8</td>
<td>40.8</td>
<td>56.3</td>
<td>81.1</td>
<td>90.0</td>
<td>116.2</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>5.4</td>
<td>19.1</td>
<td>29.9</td>
<td>35.1</td>
<td>44.9</td>
<td>51.1</td>
<td>92.9</td>
<td>∞</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>2.6</td>
<td>12.9</td>
<td>15.4</td>
<td>21.1</td>
<td>31.3</td>
<td>42.5</td>
<td>54.7</td>
<td>∞</td>
</tr>
</tbody>
</table>

ages and noise corruption scenarios, it is found, as demonstrated
in Section V, that such a decoupled optimization process can
achieve a superior performance while maintaining a competitive
computation efficiency.

Table II lists the threshold set obtained by the proposed
optimization process using a 3 × 3 filter window \((N = 9)\). The
parameter \(K \triangleq \frac{N}{2} = 4\), which is the maximum valid cen-
tral weight of the CWTVM filter for the given filter window. The
parameter \(L\) is preset to \((N - 1) = 8\), so that the clas-
sifier has \((N - 1)\) decision levels to match each reference esti-
mate to the signal characteristics of all pixels in the window.
Such a configuration leads to a total number of partition cells
\(M = L^K = 4096\). This partition threshold set is used in all
our experiments without any further optimization. It is noted
that other threshold values \((\tilde{T}_{\lambda,k})\)s or a different partition cell
number \((M)\) may be available for the proposed PBTVM filter
with a 3 × 3 filter window, and the partition threshold set for the
PBTVM filter with a 5 × 5 filter window size can be obtained
using the proposed formulation technique as well. Neverthe-
less, all the experiments in Section V suggest that the PBTVM
filter with the partition threshold set of Table II and a 3 × 3 filter
window, if implemented recursively, is able to achieve satisfac-
tory performance under most of noise corruption conditions.

B. Constrained LMS Weight Training

For a given filter window \(W(c)\), the filtering weight set,
\(w_i\), which relates to the classified structure partition cell \(\Psi_i\), is
trained to minimize the mean-square error (MSE) given by

\[ J = \mathbb{E}[[s(c) - \hat{x}(c)]^2] \]  

(20)

where \(\mathbb{E}[\cdot]\) is the expectation operator, \(s(c)\) and \(\hat{x}(c)\) denote,
respectively, the original and the reconstructed pixel value by
the PBTVM filter at coordinate \(c\). Since all the partition cells, \{\(\Psi_i\ |
\ i = 1, 2, \ldots, M\}\}, are mutually exclusive to each other, the
minimization of the overall MSE can be equivalently achieved by
the independent MSE minimization with respect to each parti-
cell \(\Psi_i\) over the entire test image, that is

\[ J_i = \mathbb{E}[[s(c) - \hat{x}(c)]^2 | p(c) = i] \]  

(21)

where \(p(c)\) is the partition index defined by (13). Following the
steps similar to [39] and [40], the updating procedure for the
weight coefficients, \(w_i = [w_{i,0}, w_{i,1}]^T\), at the \(n\)th iteration is
given by

\[ w_{i,n+1} = w_{i,n} + 2\lambda_n (I \otimes [s(c) - \hat{x}(c)]D(c)) \]  

(22)

\[ u_{i,n+1} = 1 - 1^T w_{i,n+1} \]  

(23)

where \(i = p(c)\) is the index to the classified partition cell, \(I\) is
a \(K \times K\) identity matrix, \(\otimes\) denotes the Kronecker (or Tensor)
product [49], \(1 = [1, 1, \ldots, 1]^T\) is a \(K \times 1\) unitary vector, and

\[ D(c) = \begin{bmatrix} d_1(c) \\ d_2(c) \\ \vdots \\ d_K(c) \end{bmatrix} = \hat{Y}(c) - (1 \otimes x(c)) \]  

(24)

is a column vector consisting of all the difference vectors be-
tween the CP and the reference estimates from the \(\hat{Y}(c)\) of (19),
so that each element of \(D(c)\) can be expressed as \(d_k(c) =
\hat{y}_k(c) - \hat{x}(c)\) for \(1 \leq k \leq K\).

In order to obtain a steady convergence in a nonstationary en-
vironment such as image processing applications, the step-size parameter in (22) has to be adaptive to and, therefore, normal-
ized by the input energy [39]. As a result, the parameter \(\lambda_n\) at
the \(n\)th iteration is given by

\[ \lambda_n = \frac{\lambda_0}{\|D(c)\|^2} = \frac{\lambda_0}{\sum_{k=1}^K \|d_k(c)\|^2} \]  

(25)

where \(0 \leq \lambda_0 \leq (2/3)\) is a constant to guarantee the sufficient
condition of a strict convergence.

Extensive tests have confirmed that the proposed weight
training algorithm can steadily converge to a desired stopping
criterion in a moderate number of iterations. Fig. 3 demonstrates the results with three 256 × 256-pixel 24-bit RGB
images, Lena, Peppers, and Parrots, corrupted by 10% random
impulse noise. Given a stopping criterion, \(\Delta\)MSE ≤ 0.01,
the training algorithm approaches a steady solution within, at
most, 60 iterations. The tests with other corruption scenarios,
although not presented in this paper, have demonstrated a very
similar convergence trend.

The dependence between the error criterion (MSE) and the
parameter \(\lambda_0\) is also evaluated through the tests. Fig. 4 shows the results with Lena, Peppers, and Parrots corrupted by 5% and
10% impulse noise, respectively, where the value of \(\lambda_0\) covers
the entire range of \((0, 2/3]\) with an increment of 0.001. The
optimal value of \(\lambda_0\) always lies in the range of \([0.002, 0.05]\).
Thus, a suboptimal value, \(\lambda_0 = 0.02\), is used throughout our
training and testing process.

C. Recursive Filtering and Weight Training

The recursive implementation of weighted nonlinear filters
has been examined in earlier published work [40], [41]. As these
filtering structures are able to adapt the weighting operation to
the input values, better noise suppression is generally achieved.
is the recursive counterpart. In other words, the filter is formulated to achieve better restoration performance. Given a filter window, $\hat{W}(c)$, containing $N$ pixel samples and denoting $K = (N - 1)/2$, the input set of the recursive PBTVM filter is given in a lexicographic scan order as follows:

$$\hat{W}(c) = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_{K-1}, \hat{x}_K, \hat{x}_{K+1}, \ldots, \hat{x}_N\} \quad (26)$$

where $\{\hat{x}_k \mid k = 1, 2, \ldots, K - 1\}$ are the outputs of the recursive PBTVM filter at previous pixel positions, and $\{x_k \mid k = K, K+1, \ldots, N\}$ are the inputs directly from the current filter window $\hat{W}(c)$. Given such an input set $\hat{W}(c)$, the implementation of the recursive PBTVM filter is very similar to its non-recursive counterpart. In other words, the $\hat{W}(c)$ is input into the CWTVM filter to produce the reference estimates through (3)–(7), then the reference estimates are fed into the structure classification via (10)–(14), and the filtering output is finally produced by the partition-based weighted operation given by (15)–(18).

Although the recursive PBTVM filter achieves a better overall performance, it precludes a closed-form analytical solution to its weight training process. This is because its input set, $\hat{W}(c)$, substantially depends on the previous outputs of the filter. A heuristic weight training strategy is developed here as an alternative, which is able to satisfy the weight training of the recursive PBTVM filter, even though many assumptions of a steady convergence do not necessarily hold. Given the weight set at the $t$th iteration as $W^t = \{w_{11}^t, w_{12}^t, \ldots, w_{M1}^t\}$, where $w_{ij}^t$ is the weight vector of the partition cell $\hat{W}_i$, and $i = 1, 2, \ldots, M$, the proposed weight training strategy is given as follows.

1) At the first iteration, the weight set is updated by the training algorithm defined in (22)–(25) and using the

![Fig. 3. Convergence evaluation of the proposed training algorithm using test images Lena, Peppers, Parrots corrupted by 10% random impulse noise. (a) MSE versus iteration. (b) Updating error versus iteration.](image1)

![Fig. 4. Dependence of the error measurement (MSE) on the training step parameter $\lambda_0$ using test images Lena, Peppers, and Parrots corrupted by 5% and 10% impulse noise, respectively. (a) 5% random impulse noise. (b) 10% random impulse noise.](image2)
nonrecursive input set $\mathbf{W}(c)$. The resultant weight set, denoted as $\mathcal{W}^A$, is then used by the PBTVM filter to produce the outputs, via (3)–(18).

2) At the second iteration, the outputs of step (1) is used to construct the input set $\tilde{\mathbf{W}}(c)$ of (26). This recursive input set is then used to update the weight set, via the training algorithm given by (22)–(25).

3) For any iteration $t > 2$, the outputs of the PBTVM filter at iteration $(t - 1)$ and the weight set $\mathcal{W}^{t-1}$ are used to construct the current recursive input set $\tilde{\mathbf{W}}(c)$. This input set is then used for further weight set updating via the training algorithm given by (22)–(25).

Fig. 5 depicts the schematic diagram of the recursive weight training strategy. Despite its intractability in mathematical analysis, the proposed recursive training strategy always converges to solutions with a performance better than that of its nonrecursive counterpart in extensive tests conducted on real image data. Fig. 6 demonstrates the convergence of the recursive weight training strategy on several test images corrupted by different degrees of impulse noise.

V. SIMULATION RESULTS

The proposed PBTVM filter has been evaluated by an extensive range of tests, and its performance is compared with a number of prior-art filtering techniques in the area of color image restoration. The tests and analysis are summarized as follows:

1) a performance comparison of the PBTVM filter with other prior-art impulse filtering techniques with respect to different levels of impulse corruptions;
2) a robustness assessment of the PBTVM filter with respect to the image or noise characteristics variation in its weight training and filtering operation;
3) a performance comparison of the PBTVM filter with other state-of-the-art adaptive filters in suppressing additive noise or mixed noise corruption; and
4) a computational complexity analysis of the PBTVM filter framework.

Several objective criteria are used in the tests to measure the distortion in filtering outputs, which include the MSE, the mean absolute error (MAE), and the normalized color difference (NCD). The MSE and MAE are defined in the RGB color space and given as follows [12], [14]:

\[
\text{MSE} = \frac{1}{mWH} \sum_{c \in C} ||\mathbf{s}(c) - \tilde{\mathbf{x}}(c)||_2^2 \tag{27}
\]

\[
\text{MAE} = \frac{1}{mWH} \sum_{c \in C} ||\mathbf{s}(c) - \tilde{\mathbf{x}}(c)||_1 \tag{28}
\]

where $\mathbf{s}(c)$ and $\tilde{\mathbf{x}}(c)$ are, respectively, the original and the restored pixel at pixel coordinate $c \in C; W$ and $H$ denote the width and the height of the test image, respectively; and $m$ represents the number of color channels. Please note that $|| \cdot ||_2$ denotes the $l_2$ norm (Euclidean distance) and $|| \cdot ||_1$ denotes the
where for a given pixel coordinate \( c \in C \) in the CIELUV color space, \( E_{\text{LUV}}(c) \) and \( \Delta E_{\text{LUV}}(c) \) represent, respectively, the magnitude of the original pixel, and the difference between the original pixel and its reconstruction.

### A. Impulse Noise Suppression

In terms of the impulse noise suppression, the impulse noise model proposed by [12], [14], [16] is used in the tests to simulate the impulse corruption present in natural color images, which is given by

\[
x(c) = \begin{cases} 
    v(c), & \text{with probability } p_v \\
    s(c), & \text{with probability } 1 - p_v
\end{cases}
\]

(30)

where \( s(c) \) and \( x(c) \) denote, respectively, the original pixel and the sample pixel at a pixel coordinate \( c \). \( p_v \) denotes the noise ratio (the percentage of the corrupted pixels in a given image), and \( v(c) \) is a noisy version of \( s(c) \), who suffers from impulse corruption to, at least, one of its RGB components. For 24-bit RGB test images, when the impulse amplitudes take on the value of 0 or 255 with an equal probability, the impulse noise is known as the pepper-and-salt (PS) impulse. However, if the impulse amplitudes are distributed randomly in the range of \([0, 255]\), a more generalized type of impulse noise is generated and named as the random impulse noise. Such a random impulse noise is used in this section for all the tests.

All the impulse corruptions are generated according to the model in (30), using the random impulse noise and a noise ratio \( p_v \), varied from 0% to 30%. A two-step process [8], [26], [47] is applied to simulate the channel correlation of the impulse corruption. Given the preset noise ratio \( p_v \), each RGB component of the test image is corrupted independently by the random impulse noise. Then a factor \( p = 0.5 \) is used to simulate the channel correlation for each corrupted pixel, that is, if a pixel is corrupted by random impulses on at least one of its components, then its noise-free components have a 50% probability to be corrupted, as well.

The evaluations of impulse suppression are conducted on three 256 × 256-pixel 24-bit RGB images, Lena, Peppers, and Parrots, which have been widely used by prior-art impulse filtering techniques [12]–[16] for their representative color characteristics and image structures. Two classes of vector filters are involved in the impulse suppression tests. The first includes classic vector filters such as the VMF [7], the generalized vector directional filter (GVDF) [8], and the DDF [9]. The second is the state-of-the-art techniques recently developed for the impulse suppression, which includes the self-adaptive algorithm (SAA) [19], the AVMF [12], the AVLUM [13], the histogram-based trimmed median (HBTM) filter [22], the SCWVDF/ACWVDF [14], and the selection weighted vector directional filter (SWVDF) [15]. The tested filters are implemented with a 3 × 3-pixel window in all the tests, which slide from pixel to pixel in a raster scanning fashion.

**Experiment 1—Performance Within the Training Set:** This experiment is designed to assess the performance of the vector filters aforementioned in restoring different levels of impulse corruptions. Parameter settings as proposed in the original works are used by the AVLUM, the HBTM, the AVMF, the ACWVDF/SCWVDF during the tests. The noise ratio estimator is bypassed in the SAA filter and the actual noise ratio is used instead. The proposed PBTVM filter is applied within the training set, that is, the same image and noise ratio to be filtered are used in the off-line weight training of the filter.

Figs. 7–9 present the comparative results of restoring the test images Lena, Peppers, and Parrots corrupted by random impulse varied from 0% to 20%. The selected noise range enables a critical examination of the detail-preserving performance on images corrupted by low noise intensities, while still allowing a sufficient noise level for the performance consistency evaluation. The prior-art ACWVDF/SCWVDF performs well at low noise ratios. However, its performance is degraded dramatically when the corruption level is increased, and is even inferior to classic vector filters such as the VMF or the DDF for serious noise corruptions with \( p_v \geq 15\% \). In contrast, the PBTVM filter has demonstrated a quite consistent improvement over other filters in a wide range of noise ratios and image structures. The improvement is most significant in Fig. 7 in terms of the MSE criterion and with the image Parrots which contains high details and specific spectral characteristics. In Figs. 7–9, where the MAE and NCD criteria are used, the PBTVM filter shows a very noticeable improvement over other filters in most cases. Please note that the results with the VMF and the DDF are not shown in Figs. 7–9 because their MSE/NCD ranges are well beyond the display scales of the figures. The restoration results for noise corruptions with \( p_v > 20\% \), although not shown in the current figures, have shown a very similar trend in the experiment.

**Experiment 2—Performance Outside the Training Set:** The robustness of a given filter in processing corrupted images outside of its training set is very important for any real restoration application. In order to assess the robustness of the PBTVM filter, three specific types of outside training modes are formulated as follows.

1) **Type-a:** The same test image but different noise corruptions are used in the weight training and the filtering operation.

2) **Type-b:** Different test images but the same noise corruption are used in the weight training and the filtering operation.

3) **Type-c:** Different test images and different noise corruptions are used in the weight training and the filtering operation.

Table III presents the experiment results on two test images, Peppers and Parrots, where the performance of the PBTVM filter working in all three outside training modes are evaluated. For Type-a PBTVM filter (PBTVM-a), the filtering weights are trained for a 15% random impulse corruption on the image to be restored in the filtering operation. For Type-b PBTVM filter (PBTVM-b), the filtering weights are trained using a different test image, Lena, corrupted by the same noise as that used in the filtering operation. The filtering weights in Type-c PBTVM filter
(PBTVM-c) are trained using the test image Lena with a 15% random impulse corruption. Performance of the most relevant state-of-the-art techniques, which include the SAA filter (with the actual noise ratio), the SWVDF (with parameter $p = 0.5$), trained using the test image Lena and a 10% random impulse corruption), and other filters with their recommended parameter settings, are also listed in the table for a comparison. Moreover, the results of the PBTVM filter with ideal training (or within the training set as mentioned previously) are also attached in the last row of the table for readers’ reference.
An important observation from Table III is that the PBTVM filter works quite robustly throughout all the corruption scenarios, despite the fact that the filtering weights used to restore the corrupted images are actually outside the training set. In most of the cases, the PBTVM filter in all three modes has produced noticeable improvements over other filtering techniques in terms of most, if not all, of performance criteria and measures. Negligible performance degradations are observed between the PBTVM filter in three test modes and the filter by the ideal training. The PBTVM filter in Type-\(a\) mode has, in general, achieved the best performance over other two modes. Even the PBTVM filter in Type-\(c\) mode, where both the training image and the training corruption are different from the corrupted images to be restored, still achieves a better performance in most scenarios in comparison with other vector filters. Extensive performance evaluations and results with other test images and noise contamination scenarios, although not shown in this paper, have reached a very similar conclusion.

Besides the objective performance improvements, the proposed PBTVM filter also achieves a consistently better perceptual quality than other impulse filtering techniques. Fig. 10 demonstrates the reconstructions of several state-of-the-art filtering techniques for the image Parrots, where a 5\% random impulse corruption is selected to reveal the detail preservation capability of the filters with respect to low noise intensity corruptions, and the filtering weights of the PBTVM filter are trained off-line on image Lena corrupted by a 15\% random impulse. It is observed that the PBTVM filter provides an almost noise free reconstruction, whilst the SCWVDF and the SAA filter still have noticeable impulses left in the background area between two birds. Moreover, the reconstruction of the PBTVM filter has well-preserved stripe structures around the eye area, which is almost as the same as, or even better than those by the SCWVDF and the SAA techniques. The subjective quality of the new filter can be further improved if a more similar image or noise corruption model is used in the weight training process.

B. Restoration of Other Noise Contamination

In addition to its superior performance in impulse noise rejection, the new filter also works satisfactorily in suppressing the Gaussian noise as well as the mixed Gaussian and impulse noise. The noise models defined in [6], [30], [47] are used in the experiments to simulate corruptions. The Gaussian noise contamination is simulated by adding in each component of image pixels an random noise which follows a zero mean Gaussian distribution and a preset standard deviation, \(\sigma\). The mixed noise contamination is simulated by a Gaussian noise contamination followed by a random impulse corruption of (30). A group of prior-art techniques for the Gaussian and mixed noise restorations have been involved in the performance comparison, which include the adaptive hybrid multivariate (AHM) filter [4], the ANNF [17], the AVDF [18], and the fuzzy vector median filter (FVMF) [26]. Classic vector filters, such as the VMF [7] and the GVDF [8], are also included in the test for references and benchmarks. A 3 \(\times\) 3 filter window size has been used in all the cases.

Table IV summarizes the results on two 256 \(\times\) 256-pixel RGB test images, Peppers and Parrots. Four corruption scenarios are listed in the table, where G10 and G20 denote two types of the Gaussian noise contaminations, respectively, with the standard deviation \(\sigma = 10\) and \(\sigma = 20\), whilst G10I2 and G20I5 represent two types of the mixed noise contaminations, the former being mixed by the Gaussian noise (\(\sigma = 10\)) and the random impulse noise (\(p_v = 2\%\)), the latter by the Gaussian noise.

---

Fig. 9. Performance of the filters in terms of the NCD, using different test images and impulse noise ratios. (a) Lena. (b) Peppers. (c) Parrots.
TABLE III
IMPULSE SUPPRESSION PERFORMANCE OF THE NEW FILTER COMPARED WITH OTHER TECHNIQUES, WHERE THE PBTVM FILTER, WORKING OUTSIDE THE TRAINING SET AND WITH A 3 × 3 WINDOW SIZE, IS USED. (a) COLOR IMAGE PEPPERS CORRUPTED BY DIFFERENT LEVELS OF RANDOM IMPULSE. (b) COLOR IMAGE PARROTS CORRUPTED BY DIFFERENT LEVELS OF RANDOM IMPULSE

(a)

<table>
<thead>
<tr>
<th>Filters</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
<td>MAE</td>
<td>NCD</td>
</tr>
<tr>
<td>SAA [19]</td>
<td>17.5</td>
<td>0.511</td>
<td>0.0045</td>
</tr>
<tr>
<td>AVLUM [13]</td>
<td>22.1</td>
<td>0.465</td>
<td>0.0042</td>
</tr>
<tr>
<td>HBTM [22]</td>
<td>19.3</td>
<td>0.466</td>
<td>0.0039</td>
</tr>
<tr>
<td>AVMF [12]</td>
<td>25.6</td>
<td>0.556</td>
<td>0.0045</td>
</tr>
<tr>
<td>ACWVDF [14]</td>
<td>21.1</td>
<td>0.693</td>
<td>0.0072</td>
</tr>
<tr>
<td>SCWVDF [14]</td>
<td>20.5</td>
<td>0.666</td>
<td>0.0070</td>
</tr>
<tr>
<td>SWVDF [15]</td>
<td>18.1</td>
<td>0.962</td>
<td>0.0142</td>
</tr>
<tr>
<td>PBTVM-a</td>
<td>14.8</td>
<td>0.472</td>
<td>0.0045</td>
</tr>
<tr>
<td>PBTVM-b</td>
<td>14.1</td>
<td>0.432</td>
<td>0.0039</td>
</tr>
<tr>
<td>PBTVM-c</td>
<td>16.8</td>
<td>0.476</td>
<td>0.0044</td>
</tr>
<tr>
<td>PBTVM</td>
<td>11.0</td>
<td>0.381</td>
<td>0.0038</td>
</tr>
</tbody>
</table>

(b)

<table>
<thead>
<tr>
<th>Filters</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSE</td>
<td>MAE</td>
<td>NCD</td>
</tr>
<tr>
<td>SAA [19]</td>
<td>23.5</td>
<td>0.426</td>
<td>0.0025</td>
</tr>
<tr>
<td>AVLUM [13]</td>
<td>25.3</td>
<td>0.502</td>
<td>0.0027</td>
</tr>
<tr>
<td>HBTM [22]</td>
<td>28.3</td>
<td>0.570</td>
<td>0.0027</td>
</tr>
<tr>
<td>AVMF [12]</td>
<td>36.5</td>
<td>0.746</td>
<td>0.0037</td>
</tr>
<tr>
<td>ACWVDF [14]</td>
<td>24.8</td>
<td>0.425</td>
<td>0.0023</td>
</tr>
<tr>
<td>SCWVDF [14]</td>
<td>19.6</td>
<td>0.377</td>
<td>0.0019</td>
</tr>
<tr>
<td>SWVDF [15]</td>
<td>18.5</td>
<td>0.745</td>
<td>0.0033</td>
</tr>
<tr>
<td>PBTVM-a</td>
<td>14.4</td>
<td>0.444</td>
<td>0.0025</td>
</tr>
<tr>
<td>PBTVM-b</td>
<td>17.2</td>
<td>0.467</td>
<td>0.0025</td>
</tr>
<tr>
<td>PBTVM-c</td>
<td>20.5</td>
<td>0.514</td>
<td>0.0027</td>
</tr>
</tbody>
</table>
| PBTVM    | 7.4     | 0.283   | 0.0020  | 14.1    | 0.526   | 0.0036  | 39.7    | 1.200   | 0.0079  

$(\sigma = 20)$ and the random impulse $(p_\nu = 5\%)$. The outside training mode is used by the PBTVM filter to assess its performance robustness regarding different images and noise distributions. Its filtering weights for the Gaussian noise suppression are trained off-line on another 256 × 256-pixel RGB test image, Lena, corrupted by a Gaussian noise with $\sigma = 20$, and the filtering weights for the mixed noise suppression are trained off-line using Lena image corrupted by the mixed noise G20I5. Table IV presents the results of the test filters in restoring color images contaminated by distinct types of noise. For the
scenarios where the noise contaminations are the same as those used in weight training, the new filter achieves a considerable improvement over other techniques in terms of all the objective criteria. For the scenarios where the training condition and filtering condition are totally different, the new filter still delivers a robust performance, with its results slightly worse than the top performers (e.g., the ANNF) but is still quite comparable to those of the rest techniques. There are a few cases where the NCD measures of the PBTVM filter are slightly worse than those of prior-art techniques, but the new filter always achieves a better subject quality than its counterparts.

Fig. 11 highlights the image Parrots restored by the PBTVM filter and other prior-art techniques from a mixed noise contamination, G2015. Blocks of color distortion remain in the output.
Table IV presents the elementary operations required for a number of vector filters, where the VMF, the SAA and the AVLUM use the Euclidian distance, the GVDF and the SCWVDF employ the vector directional distance, and the SWVDF utilizes the distance-directional distance. The weights of the SWVDF and the PBTVM filter are trained off-line, and the noise/parameter estimations of the SAA filter are bypassed for a maximum computation efficiency. The HBTM and the SAA filters occur a quite low computational cost which is comparable to the VMF. The computational complexity of the PBTVM filter is fairly reasonable despite its three-stage adaptation structure. Its operations associated with the reference estimation, structure partition, and weighted filtering are of on the pixel basis, which is quite common to any selective vector filter, such as the AVLUM or the SWVDF. However, since the arc cosine operation usually cost more computation than other operations, the computational complexity of the PBTVM filter is usually much lower than that of the SCWVDF, and slightly higher than that of the AVLUM filter.

Table VI lists the execution times of a number of filters in processing the 256 × 256 RGB image Lena corrupted by 10%

of the recursive mean filter. The AHM filter works quite well in preserving the image details (i.e., the stripes around birds’ eyes). However, some pepper-like impulses are left in the reconstruction as a result of inaccurate/inadequate local statistics estimation. The ANNF and the AVDF preserve the edge integrity, but fine structures are blurred and the background is highly distorted by chromatic blotches. A better result is achieved by the PBTVM filter. It provides a smoother chromatic background, while the fine stripes and bird eyes are clearly preserved.

C. Computational Complexity Analysis

A general framework has been used to analyze the computational requirements of image filters [15], [19], [30], [48]. The framework evaluates the computational complexity of a specific filter by counting the total elementary operations required to processing an $n_x \times n_y$ window which contains $n^2$ vector samples. Several elementary operations are frequently taken into account in the analysis, which include additions (ADDs), multiplications (MULTs), divisions (DIVs), square roots (SQRTs), comparisons (COMPs), exponents (EXPs), and arc cosines (ARCCOSs).
random impulse noise using a $3 \times 3$-pixel window. Please note that the program was written in C++ and run on the same simulation platform (P4 3.0 GHz/2096 MB DDR/Window XP Pro) without any code optimization. Two work modes of the SAA filter are tested separately, i.e., the maximum computation efficiency model (SAA-a) where both the noise estimator and the parameter estimation were bypassed, and the normal work mode (SAA-b) where both the noise estimator and the parameter estimation are activated. It is interesting to note that execution time of the PBTVM filter is just slightly longer than that of the AVLUM, while comparable to that of the SAA-b filter where the on-line parameter estimation is activated. In contrast,
TABLE V
COMPUTATIONAL COMPLEXITY ANALYSIS OF THE PROPOSED PBTVM FILTER COMPARED WITH OTHER TECHNIQUES

<table>
<thead>
<tr>
<th>Operations</th>
<th>ADDs</th>
<th>MULts/DIVs</th>
<th>SQRTs</th>
<th>COMPs</th>
<th>EXPs</th>
<th>ARCCOSs</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMF [7]</td>
<td>(O(n^4))</td>
<td>–</td>
<td>(O(n^4))</td>
<td>(O(n^2))</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>GVDF [8]</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>(O(n^2))</td>
<td>–</td>
<td>(O(n^4))</td>
</tr>
<tr>
<td>SAA [19]</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>(O(n^2))</td>
<td>(O(n^4))</td>
<td>–</td>
</tr>
<tr>
<td>HBTM [22]</td>
<td>(O(n^4))</td>
<td>(O(n^2))</td>
<td>(O(n^4))</td>
<td>(O(n^2))</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SWVDF [15]</td>
<td>(O(n^3))</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>(O(n^2))</td>
<td>(O(n^4))</td>
<td>–</td>
</tr>
<tr>
<td>AVLUM [13]</td>
<td>(O(n^4))</td>
<td>–</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SCWVDF [14]</td>
<td>(O(n^3))</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>–</td>
<td>(O(n^4))</td>
</tr>
<tr>
<td>PBTVM</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>(O(n^4))</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

TABLE VI
EXECUTION TIME OF THE PROPOSED PBTVM FILTER COMPARED WITH OTHER TECHNIQUES, WHERE THE 256 \times 256 RGB IMAGE LENA CORRUPTED BY 10% RANDOM IMPULSE NOISE IS USED, AND ALL FILTERS RAN WITH A 3 \times 3 FILTER WINDOW

<table>
<thead>
<tr>
<th>Filter</th>
<th>VMF</th>
<th>GVDF</th>
<th>AVF</th>
<th>SAA-a</th>
<th>SAA-b</th>
<th>HBTM</th>
<th>SWVDF</th>
<th>AVLUM</th>
<th>SCWVDF</th>
<th>PBTVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>2.2</td>
<td>3.4</td>
<td>2.5</td>
<td>2.5</td>
<td>11.2</td>
<td>4.4</td>
<td>8.4</td>
<td>16.4</td>
<td>10.1</td>
<td></td>
</tr>
</tbody>
</table>

\[ \begin{align*}
0 \leq e_1(c) < 20 \\
\psi_1 = 0
\end{align*} \]

\[ \begin{align*}
0 \leq e_2(c) < 15 \\
: \psi_2 = 0
\end{align*} \]

\[ \begin{align*}
15 \leq e_2(c) < \infty \\
: \psi_2 = 1
\end{align*} \]

\[ \begin{align*}
[e_1(c), e_2(c), e_3(c)]^T
\end{align*} \]

\[ \begin{align*}
0 \leq e_2(c) < 15 \\
: \psi_2 = 0
\end{align*} \]

\[ \begin{align*}
15 \leq e_2(c) < \infty \\
: \psi_2 = 1
\end{align*} \]

\[ \begin{align*}
0 \leq e_3(c) < 10 \\
: \psi_3 = 0
\end{align*} \]

\[ \begin{align*}
10 \leq e_3(c) < \infty \\
: \psi_3 = 1
\end{align*} \]

\[ \begin{align*}
0 \leq e_3(c) < 10 \\
: \psi_3 = 0
\end{align*} \]

\[ \begin{align*}
10 \leq e_3(c) < \infty \\
: \psi_3 = 1
\end{align*} \]

\[ \begin{align*}
0 \leq e_3(c) < 10 \\
: \psi_3 = 0
\end{align*} \]

\[ \begin{align*}
10 \leq e_3(c) < \infty \\
: \psi_3 = 1
\end{align*} \]

\[ \begin{align*}
0 \leq e_3(c) < 10 \\
: \psi_3 = 0
\end{align*} \]

\[ \begin{align*}
10 \leq e_3(c) < \infty \\
: \psi_3 = 1
\end{align*} \]

\[ \begin{align*}
0 \leq e_3(c) < 10 \\
: \psi_3 = 0
\end{align*} \]

\[ \begin{align*}
10 \leq e_3(c) < \infty \\
: \psi_3 = 1
\end{align*} \]

\[ \begin{align*}
0 \leq e_3(c) < 10 \\
: \psi_3 = 0
\end{align*} \]

\[ \begin{align*}
10 \leq e_3(c) < \infty \\
: \psi_3 = 1
\end{align*} \]

Fig. 12. Calculation of the partition index \(p(c)\) for the input distance vector, \(\bar{e}(c) = [e_1(c), e_2(c), e_3(c)]^T\), where the partition threshold set is given by Table VII with \(K = 3\) and \(L = 2\).
the SCWVDF, which follows a similar structure to that of the AVLUM but using the vector directional distance, requires almost 1.6 times the computational cost of the PBTVM filter to process the same corrupted image.

VI. CONCLUSION

A partition-based adaptive vector filter framework is proposed for suppressing distinct types of noise in digital color images. The filter uses a series of robust reference filters to estimate the local structure, and utilizes a structure classification to map the inputs into mutually exclusive partition cells. Adaptive weighting filtering is formulated to achieve the best restoration performance for the given local structure, where the filtering weights are optimized off-line for each partition cell through a constrained LMS algorithm. The proposed filtering technique demonstrates a superior performance over other state-of-the-art vector filters in suppressing the random impulse noise, the Gaussian noise, as well as the mixed Gaussian and impulse noise. Significant improvements have been observed in terms of standard objective measurements and perceived image quality.

APPENDIX

Example 1: Given the pixel samples of a $3 \times 3$ filter window, $W(c)$, in a lexicographic scan order as follows:

$$
\begin{align*}
\{x_{11}(c), x_{12}(c), x_{13}(c), x_{21}(c), x_{22}(c), x_{23}(c), x_{31}(c), x_{32}(c), x_{33}(c)\} = [76, 71, 51, 72, 74, 255, 71, 255, 51, 73, 73, 70, 71, 70, 65, 51].
\end{align*}
$$

where three local pixels, $x_3(c)$, $x_4(c)$, and $x_5(c)$, have been corrupted by the Salt-and-pepper impulse noise, and the central pixel (CP), $x(c) = x_{31}(c) = [74, 255, 52]^T$. The outputs of the CWVM filter with central weight $k = 1, 2, 3, 4$, respectively, are produced according to (1)–(2) and listed as follows:

$$
\begin{align*}
\{y_{11}(c), y_{12}(c), y_{13}(c), y_{14}(c)\} = [73, 75, 52, 73, 75, 52, 74, 255, 52].
\end{align*}
$$

With a low center weight value applied, the outputs of the CWVM filter, e.g., $y_1(c)$ and $y_2(c)$, are able to assemble the noise-free CP value. However, when the center weight becomes higher, the outputs of the CWVM filter, e.g., $y_3(c)$ and $y_4(c)$, suffer from serious impulse corruption.

Example 2: Consider again the filter window $W(c)$ given in Example 1, where $N = 9$ and $x(c) = x_{31}(c) = [74, 255, 52]^T$. Applying the CP-distance ranking method of (3)–(5) to the observed pixels will result in a re-ordered pixel sequence $W(c)$ as follows:

$$
\begin{align*}
\{x_{11}(c), x_{12}(c), x_{13}(c), x_{21}(c), x_{22}(c), x_{23}(c), x_{31}(c), x_{32}(c), x_{33}(c)\} = [74, 255, 52, 75, 255, 0, 76, 76, 51, 73, 75, 52, 72, 71, 49, 71, 69, 50, 70, 68, 51, 70, 65, 51, 71, 255, 255].
\end{align*}
$$

Note that three corrupted pixels, $x_3(c)$, $x_4(c)$, and $x_5(c)$ of the $W(c)$, are now ranked in the last $(x_{31}(c))$, the second $(x_{22}(c))$, and the first $(x_{11}(c))$ positions in the $W(c)$, respectively. So, when the central weight $k > 1$, one of these corrupted pixels, $x_3(c)$ (i.e., $x_{31}(c)$), will be trimmed out of the modified CW-ADV calculation (7). As a result, the outputs of the CWTVM filter with central weights, $k = 1, 2, 3, 4$, will be

$$
\begin{align*}
\{y_{11}(c), y_{12}(c), y_{13}(c), y_{14}(c)\} = [73, 75, 52, 73, 75, 52, 74, 255, 52].
\end{align*}
$$

With the contribution of the CP-distance trimming technique, the CWTVM filter with $k = 3$ is able to produce a noise free output, whilst the output of the CWCM filter with the same center weight still suffers from the impulse corruption.

Example 3: Given $K = 3$ and $L = 2$, the total number of partition cells $M = L^K = 8$. Fig. 12 demonstrates the calculation of the partition index $p(c)$ according to (13) and (14), where the input distance vector is $e(c) = [e_1(c), e_2(c), e_3(c)]^T$, and the threshold set \{$T_{k,l} : 1 \leq k \leq K, 0 \leq l \leq L$\} is given by Table VII.

ACKNOWLEDGMENT

The authors would like to thank Dr. R. Lukac of the Bell Canada Multimedia Laboratory, University of Toronto, for kindly providing all the corrupted test images, and the anonymous reviewers for their valuable comments which helped to improve the quality of the paper.

REFERENCES


Zhouhna Ma (M’02) was born in Zhejiang, China, in 1973. He received the B.Eng. degree from Shanghai Jiaotong University, China, the M.Eng. degree from the Dalian University of Technology, China, in 1995 and 1999, respectively, and the Ph.D. degree in communications and information systems from Shanghai Jiaotong University, China, in 2002. From 2002 to 2004, he was a Research Fellow conducting an Australia Research Council (ARC) funded project with the School of Computer Science and Software Engineering, Monash University, Melbourne, Australia. Since 2004, he has been a Research Fellow with the
Biomedical and Multimedia Information (BMIT) Group in School of Information Technologies, University of Sydney, Sydney, Australia. His research interests include multivariance signal processing, color image restoration, video compression, and robust streaming.

Hong Ren Wu was born in Beijing, China, in 1956. He received the B.Eng. and M.Eng. degrees from the University of Science and Technology (formerly Beijing University of Iron and Steel Technology), Beijing, in 1982 and 1985, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Wollongong, Wollongong, Australia, in 1990.

From 1982 to 1985, he was an Assistant Lecturer with the Department of Industrial Automation, University of Science and Technology. He joined the Department of Robotics and Digital Technology, Chisholm Institute of Technology, and then the Faculty of Information Technology, Monash University, Melbourne, Australia, where he served as a Lecturer (1990 to 1992), Senior Lecturer (1992 to 1996), and Associate Professor of digital systems (1997 to 2005). Since 1997, he has been with the School of Electrical and Computer Engineering, Royal Melbourne Institute of Technology, Melbourne, Australia, where he is on the academic staff as Professor of visual communications engineering and the Director of Software and Network Engineering. He has published over 300 scholarly research papers, pioneered several new research directions, and made a number of landmark contributions in his field. More important, however, is that many of his research results have been translated into solutions to real-life problems worldwide and have made tremendous improvements to the quality of life.

Dr. Feng is a Fellow of ACS, ATSE, HKIE, and IEE. He received the Crump Prize for Excellence in Medical Engineering. He is currently the Special Area Editor of the IEEE Transactions on Information Technology in Biomedicine and Chairman of IFAC-TC-BIOMED.