MRL-Filters: A General Class of Nonlinear Systems and Their Optimal Design for Image Processing

Lúcio F. C. Pessoa, *Student Member, IEEE,* and Petros Maragos, *Fellow, IEEE*

***Abstract*—In this paper, the class of morphological/rank/linear (MRL)-filters is presented as a general nonlinear tool for image processing. They consist of a linear combination between a morphological/rank filter and a linear filter. A gradient steepest descent method is proposed to optimally design these filters, using the averaged least mean squares (LMS) algorithm. The filter design is viewed as a learning process, and convergence issues are theoretically and experimentally investigated. A systematic approach is proposed to overcome the problem of nondifferentiability of the nonlinear filter component and to improve the numerical robustness of the training algorithm, which results in simple training equations. Image processing applications in system identification and image restoration are also presented, illustrating the simplicity of training MRL-filters and their effectiveness for image/signal processing.**

***Index Terms*— Adaptive filtering, image restoration, LMS algorithm, nonlinear systems, optimal filter design, system identification.**

I. Introduction

Nonlinear filters have become very important tools in signal processing, and especially in image analysis and computer vision. Linear filters can often distort important image features, such as edges. In contrast, there are many classes of nonlinear filters that are more suitable for image analysis than linear filters because they preserve edges or directly relate to important geometrical aspects of images such as their shape and size information. A broad and useful class of nonlinear systems with these possible properties is based on the framework of mathematical morphology [7], [15].

Discrete increasing morphological systems and rank or stack filters are closely related, since they can all be represented as maxima of morphological erosions [7], [8]. Despite the wide application of those nonlinear systems, very few ideas exist for their optimal design. The current four main approaches are as follows:

1) designing morphological filters[1] as a finite union of erosions [6] based on the morphological basis representation theory [7];
2) designing stack filters via threshold decomposition and linear programming [3];
3) designing morphological networks using voting logic and rank tracing learning [16];
4) designing morphological/rank filters via a gradient-based adaptive optimization [13], [14].

Approach 1 is limited to binary increasing systems. Approach 2 is limited to increasing systems processing nonnegative quantized signals. Approach 3 needs a long time to train and convergence is very complex. In contrast, approach 4 is more general, since it applies to both increasing and nonincreasing systems and to both binary and real-valued signals. One of the successful applications of this gradient-based approach was the optimal design of min–max classifiers [17], which are closely connected to morphological systems. Our work in this paper is related only to approach 4.

For various signal processing applications, it is sometimes useful to mix in the same system both nonlinear and linear filtering strategies. Thus, hybrid systems, composed by linear and nonlinear subsystems, have frequently been proposed in the research literature. For instance, L-filters [1] are linear combinations of the order statistics of the input signal; a possible generalization of this structure is the class of Lℓ-filters [9]. Finite impulse response (FIR)-median hybrid filters [4] are median operations of a fixed number of linear FIR filters applied to the input signal. The corresponding adaptive schemes [5], [10], [12] suggested the potential of adaptive hybrid systems for image processing applications.

A common characteristic in all these nonlinear hybrid filters is their ability to deal with various types of non-Gaussian noise. It is well known that linear filters can optimally suppress additive Gaussian noise. In contrast, if the signal is corrupted by additive Laplacian noise, then a median filtering is the best way to deal with it. If the noise is impulsive, then morphological systems can be very effective in reducing it. Hence, a simple natural choice to deal with combinations of Gaussian and non-Gaussian noises is to define a hybrid system that combines the behaviors of both a linear and a nonlinear filter. Given the applicability of hybrid mor-

Manuscript received October 22, 1996; revised July 8, 1997. This work supported in part by the National Science Foundation under Grant MIP-94-21677, and by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brasília, Brazil, through a Doctoral Fellowship under Grant 200.846/92-2. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Henri Maitre.

L. F. C. Pessoa is with Motorola Semiconductor Products Sector, Austin, TX 78721 USA.

P. Maragos is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA.

Publisher Item Identifier S 1057-7149(98)04374-7.

[1] The term "morphological filter" is recently used in the area of lattice-based mathematical morphology for increasing and idempotent lattice operators; however, we use this term herein to imply a general morphological operator in analogy to the terminology "rank or linear filter."

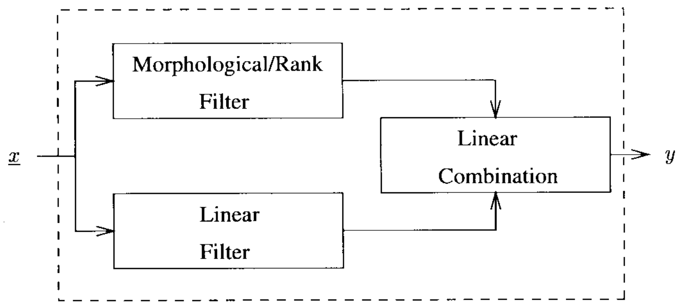


Fig. 1. Structure of the MRL-filter.

phological/rank/linear (MRL) systems and the relatively few existing ideas to design their nonlinear part, in this paper we present a general class of nonlinear systems that contains as special cases morphological, rank, and linear operators, and we develop an efficient method for its adaptive optimal design. *MRL-filters* are this class of systems, consisting of a linear combination between a morphological/rank filter and a linear FIR filter (see Fig. 1). Their nonlinear component is based on a rank function, from which the basic morphological operators of erosion and dilation can be obtained as special cases.

The contributions of this paper are:

1) extension and improvements (both theoretical and numerical) of the design algorithm proposed in [13] and [14] by using new tools such as the "rank indicator vector" for analysis and "smoothed impulses" to circumvent the nondifferentiability of rank operations;
2) theoretical conditions, with proofs, for convergence of the MRL-filter design procedure;
3) applications to image processing.

After some notation and definitions in Section II, a design methodology for MRL-filters is presented in Section III, where the filter design is viewed as a learning process, and the averaged least mean squares (LMS) algorithm is employed to adjust the filter parameters. Convergence considerations of the training process are discussed in Section IV, where important convergence conditions are proved. In Section V, we present designs of MRL-filters for problems of system identification and image restoration.

II. Preliminaries

Definition 1—Rank Function: Given a vector $\underline{t} = (t_1, t_2, \cdots, t_n)$ in $\mathbb{R}^n$, by sorting its components in decreasing order, $t_{(1)} \geq t_{(2)} \geq \cdots \geq t_{(n)}$, and picking the rth element of the sorted list, we define the rth rank function of $\underline{t}$ by

$$\mathcal{R}_r(\underline{t}) \equiv t_{(r)}, \qquad r = 1, 2, \cdots, n.$$

Definition 2—MRL-Filter: Let $\underline{x} = (x_1, x_2, \cdots, x_n)$ in $\mathbb{R}^n$ represent the values of the one-dimensional (1-D) or two-dimensional (2-D) sampled input signal (after some enumeration of the signal samples) inside an n-point moving window and let y be the output value from the filter. Then, the MRL-filter is defined as the shift-invariant system whose local signal transformation rule $\underline{x} \mapsto y$ is given by

$$\begin{aligned} y &\equiv \lambda\alpha + (1-\lambda)\beta, \\ \alpha &= \mathcal{R}_r(\underline{x} + \underline{a}) = \mathcal{R}_r(x_1 + a_1, x_2 + a_2, \cdots, x_n + a_n), \\ \beta &= \underline{x} \cdot \underline{b}' = x_1 b_1 + x_2 b_2 + \cdots + x_n b_n \end{aligned} \tag{1}$$

where $\lambda \in \mathbb{R}$, $\underline{a}$, $\underline{b} \in \mathbb{R}^n$, and "$'$" denotes transposition.

Thus, the MRL-filter is a linear combination between a morphological/rank filter and a linear filter. The vector $\underline{b} = (b_1, b_2, \cdots, b_n)$ corresponds to the coefficients of the linear FIR filter, and the vector $\underline{a} = (a_1, a_2, \cdots, a_n)$ represents the coefficients of the morphological/rank filter. We call $\underline{a}$ the "structuring element" because for $r = 1$ and $r = n$ the rank filter becomes the morphological dilation and erosion by a structuring function equal to $\pm\underline{a}$ within its support. For $1 < r < n$, we use $\underline{a}$ to generalize the standard unweighted rank operations to filters with weights, as done in [7]. The median is obtained when $r = \lfloor n/2 + 1 \rfloor$. Besides these two sets of weights, the rank r and the mixing parameter λ will also be included in the training process for the filter design. If $\lambda \in [0, 1]$, the MRL-filter becomes a convex combination of its components, so that when we increase the contribution of one component, the other one tends to decrease. For every point of the signal, we can easily see from (1) that we need $2n+1$ additions, $n+2$ multiplications and an n-point sorting operation.

Fig. 2 illustrates the usefulness of the hybrid structure of the MRL-filter for dealing with signals corrupted by non-Gaussian noise. In this example, a sinusoidal signal is corrupted either by a multivalued impulse noise[2] or by a combination of an additive Gaussian white noise and a multivalued impulse noise. Using the design procedure presented in this paper, we observed that the adaptive MRL-filter could outperform the usual flat median filter, in terms of peak-to-peak signal-to-noise ratio (PSNR), by more than 5 dB.

Due to the use of a gradient-based adaptive algorithm, derivatives of rank functions will be needed. Since these functions are not differentiable in the common sense, we will propose a simple design alternative using "rank indicator vectors" and "smoothed impulses," defined next.

Definition 3—Unit Sample Function: We define the unit sample function $q(v)$, $v \in \mathbb{R}$, as

$$q(v) \equiv \begin{cases} 1, & \text{if } v = 0 \\ 0, & \text{otherwise.} \end{cases}$$

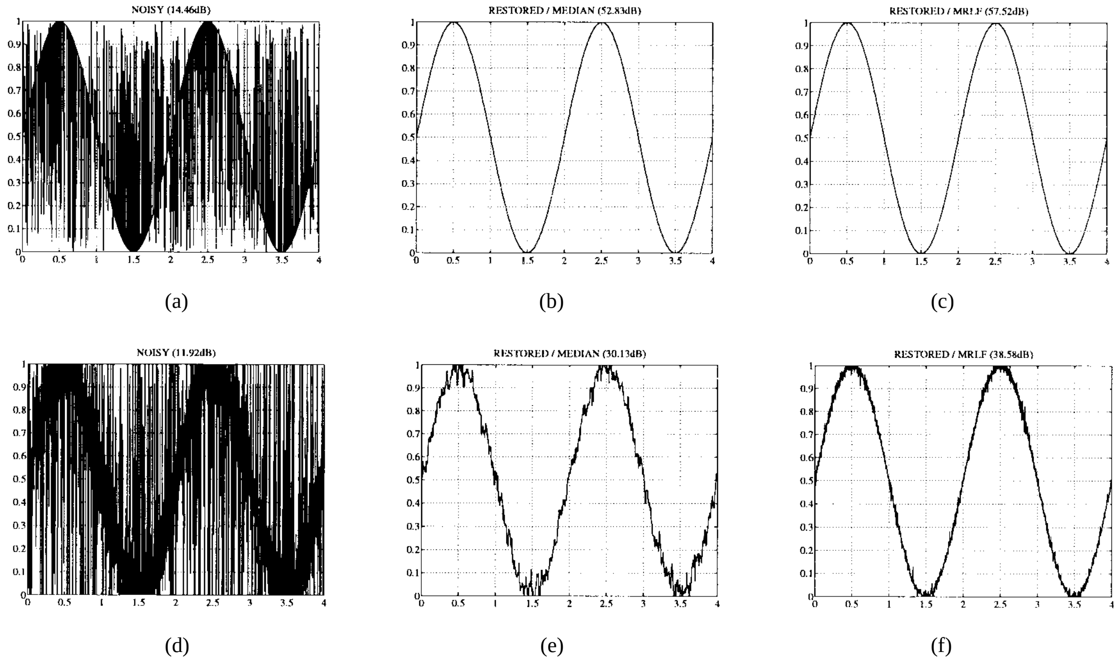
Applying q to all components of a vector $\underline{v} \in \mathbb{R}^n$, yields a vector unit sample function

$$Q(\underline{v}) \equiv [q(v_1), q(v_2), \cdots, q(v_n)].$$

Definition 4—Rank Indicator Vector: Given a vector $\underline{t} = (t_1, t_2, \cdots, t_n)$ in $\mathbb{R}^n$, and a rank $r \in \{1, 2, \cdots, n\}$, the rth rank indicator vector $\underline{c}$ of $\underline{t}$ is defined by

$$\underline{c}(\underline{t}, r) \equiv \frac{Q(z\underline{1} - \underline{t})}{Q(z\underline{1} - \underline{t}) \cdot \underline{1}'}, \qquad z = \mathcal{R}_r(\underline{t})$$

[2] The multivalued impulse noise is a generalization of the salt and pepper noise (two-valued impulse noise), where a noisy sample has a value that is uniformly distributed between the extreme values of the signal.



(a) (b) (c)

(d) (e) (f)

Fig. 2. (a) Signal corrupted by a 24% multivalued impulse noise (PSNR = 14.5 dB). (b) Signal (a) restored by a 25-point flat median filter (PSNR = 52.8 dB). (c) Signal (a) restored by a 25-point adaptive MRL-filter (PSNR = 57.5 dB). (d) Signal corrupted by a hybrid 20 dB additive Gaussian white noise and 24% multivalued impulse noise (PSNR = 11.9 dB). (e) Signal (d) restored by a 25-point flat median filter (PSNR = 30.1 dB). (f) Signal (d) restored by a 25-point adaptive MRL-filter (PSNR = 38.6 dB).

where $\underline{1} = (1, 1, \cdots, 1)$. Thus, the rank indicator vector marks the locations in $\underline{t}$ where the z value occurs; it also has unit area.

As an example, if the vector $\underline{t} = (3, 0, 5, 7, 2, 1, 3)$, then its fourth rank indicator vector is $\underline{c}(\underline{t}, 4) = \frac{1}{2}(1, 0, 0, 0, 0, 0, 1)$. Some of the properties of the rank indicator vector are summarized below.

Proposition 1: Let $\underline{t} \in \mathbb{R}^n$, $r \in \{1, 2, \cdots, n\}$, and $\underline{c} = \underline{c}(\underline{t}, r)$. Then

a) $\underline{c} \cdot \underline{1}' = 1$ (unit area);
b) $\underline{c} \cdot \underline{t}' = \mathcal{R}_r(\underline{t})$ (inner-product representation);
c) $\underline{c} \cdot (z\underline{1} - \underline{t})' = 0$, where $z = \mathcal{R}_r(\underline{t})$.
d) If r is fixed, then $\underline{c}$ is a piecewise-constant function of $\underline{t}$ with exactly $2^n - 1$ possible different values. Further, for all points $\underline{t}_0 \in \mathbb{R}^n$ with unequal components, $t_{0,\,i} \neq t_{0,\,j}\ \forall\, i \neq j$, there is a neighborhood around them such that

$$||\underline{t} - \underline{t}_0||_\infty < \tfrac{1}{2} \min_{i \neq j} |t_{0,\,i} - t_{0,\,j}|$$

inside which $\underline{c}$ is constant, i.e., $\underline{c}(\underline{t}, r) = \underline{c}(\underline{t}_0, r)$.

Proof: See the Appendix. □

Observe from Proposition 1b that the proposed MRL-filter structure is an efficient and compact way to represent a bank of linear filters. In fact, using our inner-product representation, we can write the MRL-filter as

$$y = [\lambda \underline{c} + (1 - \lambda)\underline{b},\ \lambda \underline{a} \cdot \underline{c}'] \cdot [\underline{x},\ 1]'$$

which from Proposition 1d represents a bank of $2^n - 1$ linear filters.

Proposition 2: Let $\underline{c} = \underline{c}(\underline{t}, r)$. For r fixed, if $\underline{c}$ is constant in a neighborhood of some $\underline{t}_0$, then the rth rank function $\mathcal{R}_r(\underline{t})$ is differentiable at $\underline{t}_0$ and

$$\left.\frac{\partial \mathcal{R}_r(\underline{t})}{\partial \underline{t}}\right|_{\underline{t} = \underline{t}_0} = \underline{c}(\underline{t}_0, r).$$

At points in whose neighborhood $\underline{c}$ is not constant, the rank function is not differentiable.

Proof: See the Appendix. □

At points where the function $z = \mathcal{R}_r(\underline{t})$ is not differentiable, a possible *design choice* is to assign the vector $\underline{c}$ as a one-sided value of the discontinuous $\partial z / \partial \underline{t}$. Further, since the rank indicator vector will be used to estimate derivatives and it is based on the discontinuous unit sample function, a simple approach to avoid abrupt changes and achieve numerical robustness is to replace the unit sample function by smoothed impulses, $q_\sigma(v)$, that depend on a scale parameter $\sigma \geq 0$ and have at least the following required properties:

$$\begin{aligned} q_\sigma(v) &= q_\sigma(-v) && \text{(symmetry)} \\ q_\sigma(v) &\rightarrow q(v)\ \forall\, v && \text{as } \sigma \rightarrow 0, \\ q_\sigma(v) &\rightarrow 1\ \forall\, v && \text{as } \sigma \rightarrow \infty. \end{aligned} \tag{2}$$

Functions like $\exp[-\frac{1}{2}(v/\sigma)^2]$ or $\operatorname{sech}^2(v/\sigma)$ are natural choices for $q_\sigma(v)$. Similar to $Q(\underline{v})$, the vector form of the smoothed unit sample function is

$$Q_\sigma(v) \equiv [q_\sigma(v_1),\ q_\sigma(v_2),\ \cdots,\ q_\sigma(v_n)].$$

Thus, Proposition 1b and the above considerations lead to the following definition.

Definition 5—Smoothed Rank Function: If $q_\sigma(v)$ is a smoothed impulse satisfying (2), then the smoothed rth rank function is defined as

$$\mathcal{R}_{r,\sigma}(\underline{t}) \equiv \underline{c}_\sigma \cdot \underline{t}'$$

where

$$\underline{c}_\sigma(\underline{t}, r) \equiv \frac{Q_\sigma(z\underline{1} - \underline{t})}{Q_\sigma(z\underline{1} - \underline{t}) \cdot \underline{1}'}, \qquad z = \mathcal{R}_r(\underline{t}).$$

In Definition 5, $\underline{c}_\sigma$ is an approximation for the rank indicator vector $\underline{c}$. Using ideas of fuzzy set theory, $\underline{c}_\sigma$ can also be interpreted as a membership function vector. For instance, if $\underline{t} = (3, 0, 5, 7, 2, 1, 3)$, $q_\sigma(v) = \text{sech}^2(v/\sigma)$, and $\sigma = 0.5$, then

$$\underline{c}_\sigma(\underline{t}, 4) = \tfrac{1}{2}(0.9646, 0, 0.0013, 0, 0.0682, 0.0013, 0.9646),$$

whereas $\underline{c}(\underline{t}, 4) = \frac{1}{2}(1, 0, 0, 0, 0, 0, 1)$. Some properties of $\underline{c}_\sigma$ and $\mathcal{R}_{r,\sigma}(\underline{t})$ are given in the following proposition.

Proposition 3: If $z = \mathcal{R}_r(\underline{t})$ and $z_\sigma = \mathcal{R}_{r,\sigma}(\underline{t})$, then for all $\underline{t}$ and r:

a) $\underline{c}_\sigma \cdot \underline{1}' = 1$ (unit area);
b) $\lim_{\sigma \to 0} \underline{c}_\sigma = \underline{c}$;
c) $\lim_{\sigma \to 0} z_\sigma = z$;
d) $\lim_{\sigma \to \infty} \underline{c}_\sigma = \underline{1}/n$;
e) $\lim_{\sigma \to \infty} z_\sigma = (1/n) \sum_{j=1}^{n} t_j$.

Proof: See the Appendix. □

Fig. 3 shows the smoothed rank function for different values of σ, when $\underline{t} = (t_1, t_2)$, $z = \mathcal{R}_2(\underline{t}) = \min\{t_1, t_2\}$, $z_\sigma = \mathcal{R}_{2,\sigma}(\underline{t})$, and $q_\sigma(v) = \text{sech}^2(v/\sigma)$. Observe the smoothness of z_σ for $\sigma \neq 0$, that illustrates the usefulness of smoothing impulses to circumvent the nondifferentiability of $\mathcal{R}_r(\underline{t})$.

III. Adaptive Design

From the filter definition (1), we see that our design goal is to specify a set of parameters $\underline{a}$, $\underline{b}$, r, and λ in such a way that some design requirement is met. However, instead of using the integer rank parameter r directly in the training equations, we work with a real variable ρ implicitly defined via the following rescaling[3]

$$r \equiv \left\lfloor n - \frac{n-1}{1 + \exp(-\rho)} + 0.5 \right\rfloor, \qquad \rho \in \mathrm{I\!R} \tag{3}$$

where n is the dimension of the input signal vector $\underline{x}$ inside the moving window. The relation between ρ and the output y will be defined later. In this way, the weight vector to be used in the filter design task is defined by

$$\underline{w} \equiv (\underline{a}, \rho, \underline{b}, \lambda) \tag{4}$$

but any of its components may be fixed during the process.

Our framework for adaptive design is related to adaptive filtering, where the design is viewed as a learning process and the filter parameters are iteratively adapted until convergence is achieved. The usual approach to adaptively adjust the vector $\underline{w}$, and therefore design the filter, is to define a cost function $J(\underline{w})$, estimate its gradient $\nabla J(\underline{w})$, and update $\underline{w}$ by the iterative (recursive) formula

$$\underline{w}(i+1) = \underline{w}(i) - \mu_0 \nabla J(\underline{w})|_{\underline{w}=\underline{w}(i)}, \qquad \mu_0 > 0,\ i = 0, 1, 2, \cdots \tag{5}$$

so that the value of the cost function tends to decrease at each step. The positive constant μ_0 is usually called the step size and regulates the tradeoff between stability and speed of convergence of the iterative procedure. The iteration (5) starts with an initial guess $\underline{w}(0)$ and is terminated when some desired condition is reached. This approach is commonly known as the *method of steepest descent.*

As cost function J, for the ith update $\underline{w}(i)$ of the weight vector, we use

$$J[\underline{w}(i)] = \frac{1}{M} \sum_{k=i-M+1}^{i} e^2(k) \tag{6}$$

where $M = 1, 2, \cdots$ is a memory parameter, and the instantaneous error

$$e(k) = d(k) - y(k) \tag{7}$$

is the difference between the desired output signal $d(k)$ and the actual filter output $y(k)$ for the training sample k. The memory parameter M controls the smoothness of the updating process. If we are processing noiseless signals, it is sometimes better to simply set $M = 1$ (minimum computational complexity). On the other hand, if we are processing noisy signals, we should use $M > 1$ and sufficiently large to reduce the noise influence during the training process. Further, it is possible to make a training process convergent by using a larger value of M.

Hence, the resulting adaptation algorithm, called the averaged LMS algorithm [2], is

$$\underline{w}(i+1) = \underline{w}(i) + \frac{\mu}{M} \sum_{k=i-M+1}^{i} e(k) \left. \frac{\partial y(k)}{\partial \underline{w}} \right|_{\underline{w}=\underline{w}(i)}, \qquad i = 0, 1, 2, \cdots \tag{8}$$

where $\mu = 2\mu_0$. From (1) and (4)

$$\begin{aligned} \frac{\partial y}{\partial \underline{w}} &= \left(\frac{\partial y}{\partial \underline{a}}, \frac{\partial y}{\partial \rho}, \frac{\partial y}{\partial \underline{b}}, \frac{\partial y}{\partial \lambda} \right) \\ &= \left[\lambda \frac{\partial \alpha}{\partial \underline{a}}, \lambda \frac{\partial \alpha}{\partial \rho}, (1-\lambda)\underline{x}, \alpha - \beta \right]. \end{aligned} \tag{9}$$

According to Proposition 2 and our design choice, we set

$$\frac{\partial \alpha}{\partial \underline{a}} = \underline{c} = \frac{Q(\alpha\underline{1} - \underline{x} - \underline{a})}{Q(\alpha\underline{1} - \underline{x} - \underline{a}) \cdot \underline{1}'}, \qquad \alpha = \mathcal{R}_r(\underline{x} + \underline{a}). \tag{10}$$

The final unknown is $s = \partial\alpha/\partial\rho$, which will be one more design choice. Notice from (1) and (3) that $s \geq 0$. If all the elements of $\underline{t} = \underline{x} + \underline{a}$ are identical, then the rank r does not play any role, so that $s = 0$ whenever this happens. On the other hand, if only one element of $\underline{t}$ is equal to α, then variations in the rank r can drastically modify the output α; in this case s should assume a maximum value. Thus, a possible simple choice for s is

$$\frac{\partial \alpha}{\partial \rho} = s \equiv 1 - \frac{1}{n} Q(\alpha\underline{1} - \underline{x} - \underline{a}) \cdot \underline{1}', \qquad \alpha = \mathcal{R}_r(\underline{x} + \underline{a}) \tag{11}$$

where n is the dimension of $\underline{x}$.

[3] $\lfloor \cdot \rfloor$ denotes the usual truncation operation, so that $\lfloor \cdot + 0.5 \rfloor$ is the usual rounding operation.

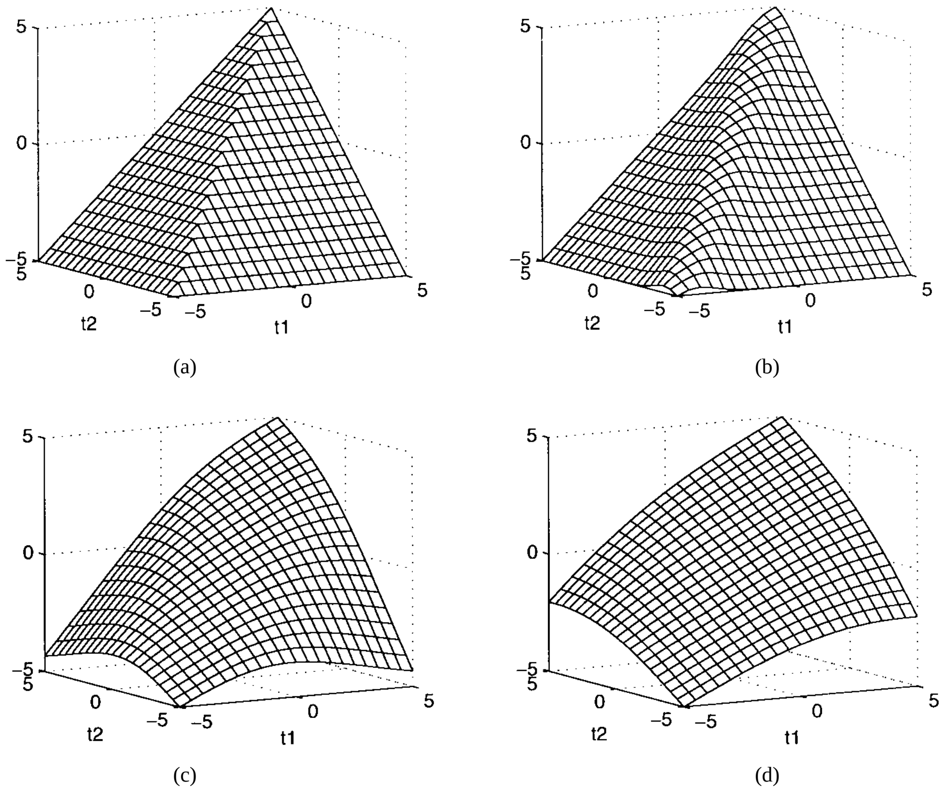


Fig. 3. Smoothed rank function with (a) $\sigma = 0$, (b) $\sigma = 1$, (c) $\sigma = 5$, and (d) $\sigma = 10$.

Finally, to improve the numerical robustness of the training algorithm, we will frequently replace the unit sample function by smoothed impulses [obeying (2)], in which case an appropriate smoothing parameter σ should be selected. A natural choice of a smoothed impulse is $q_\sigma(v) = \exp[-\frac{1}{2}(v/\sigma)^2]$, $\sigma > 0$. The choice of this nonlinearity will affect only the gradient estimation step in the design procedure (8). We should use small values of σ such that $q_\sigma(v)$ is close enough to $q(v)$. In practical applications, with sufficiently small values of σ, this transform is image independent. A possible systematic way to select the smoothing parameter σ when we choose $q_\sigma(v) = \exp[-\frac{1}{2}(v/\sigma)^2]$ could be $|q_\sigma(v)| \leq \epsilon$ for $|v| \geq \delta$, so that, for some desired ϵ and δ, $\sigma = \delta/\sqrt{\ln(1/\epsilon^2)}$. For example, if $\delta = \epsilon = 0.1$ then $\sigma \approx 0.05$. But independent of this, the proposed training algorithm works properly even with $\sigma = 0$ (no smoothing). A formal support for this is given in the next section.

Observe the simplicity of the design methodology just proposed, where the main design choices are defined in (3), (10), and (11). Since rank functions are not differentiable, optimality of their gradient estimates is not a well defined problem. The fundamental issue, instead, is how to circumvent the nondifferentiability using robust and systematic techniques. Our method, based on three design choices and a final smoothing of unit sample functions, is an efficient and simple alternative for doing that. The design choice (10) is the natural estimate to $\partial\alpha/\partial\underline{a}$ due to Propositions 2 and 3b (see Fig. 3 for a practical motivation). The design choice (11) is based on heuristic arguments, which is conveniently expressed in terms of unit sample functions. The design choice (3) is just a simple way to map from a variable $\rho \in \mathbb{R}$ to an integer rank $r \in \{1, 2, \cdots, n\}$. For example, if $\rho \rightarrow -\infty$, then $r \rightarrow n$, corresponding to a minimum operation; if $\rho \rightarrow \infty$, then $r \rightarrow 1$, corresponding to a maximum operation; if $\rho = 0$, then $r = \lfloor n/2 + 1 \rfloor$, corresponding to a median operation. The evaluation of $Q_\sigma(\alpha\underline{1} - \underline{x} - \underline{a})$ represents the major computation during the gradient estimates of the nonlinear filter component.

IV. Convergence Conditions

Some theoretical conditions for convergence of the training process (8) are presented in this section. The goal is to find upper bounds μ_w to the step size μ, such that (8) can converge if $0 < \mu < \mu_w$. For the sake of simplicity, we assume the framework of system identification with noiseless signals, and consider the training process of only one element of $\underline{w}$ at a time, while the others are optimally fixed. This means that given the original and transformed signals, and three parameters (sets) of the original $\underline{w}^* = (\underline{a}^*, \rho^*, \underline{b}^*, \lambda^*)$ used to transform the input signal, we will use (8) to track only the fourth unknown parameter (set) of $\underline{w}^*$ in a noiseless environment. If the training process (8) is convergent, then $\lim_{i\rightarrow\infty} ||\underline{w}(i) - \underline{w}^*|| = 0$, where $|| \cdot ||$ is some error norm. By analyzing the behavior of $||\underline{w}(i) - \underline{w}^*||$, under the above assumptions, we have found the following conditions for convergence.

Proposition 4: A necessary condition for $\lim_{i\rightarrow\infty} ||\underline{a}(i) - \underline{a}^*|| = 0$ in the training process (8) with any bounded initial condition $\underline{a}(0)$, given that $\rho(i) = \rho^*$, $\underline{b}(i) = \underline{b}^*$, and $\lambda(i) = \lambda^*$

$\forall i,\ \lambda^* \neq 0$, is

$$0 < \mu < \mu_a \equiv 2M\lambda^{-2}.$$

Proof: See the Appendix. □

Observe the interesting fact that the above condition for training the coefficients of the morphological/rank part of the filter is not dependent on the input data. On the other hand, this is not the case for the linear part of the filter, as will be seen later. The theoretical upper bound μ_a may be too large for practical implementations; thus, in some applications a smaller practical upper bound may often suffice for convergence.

Proposition 5: A sufficient condition for $\lim_{i\to\infty} |r(i) - r^*| = 0$ and $\lim_{i\to\infty} |\rho(i+1) - \rho(i)| = 0$ in the training process (8) with any bounded initial condition $\rho(0)$, given that $\underline{a}(i) = \underline{a}^*$, $\underline{b}(i) = \underline{b}^*$, and $\lambda(i) = \lambda^*\ \forall i,\ \lambda^* \neq 0$, is

$$0 < \mu < \mu_\rho \equiv \frac{\lambda^{-2}}{(n-1)D}$$

where $D = \max_{k,l}\{|x(k) - x(l)|\} + \max_{k,l}\{|a_k - a_l|\}$.

Proof: See the Appendix. □

The theoretical upper bound μ_ρ may be too small, but guarantees convergence. Due to the discrete nature of the rank parameter r, very fast convergence will usually be observed. Notice that, although the actual final objective in this case is to track the optimal integer rank r^*, its corresponding from (3) real parameter ρ^* is not unique; i.e.,

$$\ln\left(\frac{n - r^* - 0.5}{r^* - 0.5}\right) < \rho^* < \ln\left(\frac{n - r^* + 0.5}{r^* - 1.5}\right).$$

Proposition 6: A necessary condition for $\lim_{i\to\infty} \|\underline{b}(i) - \underline{b}^*\| = 0$ in the training process (8) with any bounded initial condition $\underline{b}(0)$, given that $\underline{a}(i) = \underline{a}^*$, $\rho(i) = \rho^*$, and $\lambda(i) = \lambda^*$ $\forall i,\ \lambda^* \neq 1$, is

$$0 < \mu < \mu_b \equiv \frac{2M}{n}\,[L(1-\lambda)]^{-2}$$

where $L = \max_k\{|x(k)|\}$.

Proof: See the Appendix. □

Notice that, contrary to the case of the morphological/rank part of the filter, the input data has a crucial influence on the maximum step size in the case of its linear part. The condition in Proposition 6 will frequently require quite small values of step size, therefore contributing to slow convergence rates. In this sense, designing the nonlinear component of the MRL-filter is generally more effective than designing its linear component. This fact is demonstrated in Section V.

Proposition 7: A necessary condition for $\lim_{i\to\infty} |\lambda(i) - \lambda^*| = 0$ in the training process (8) with any bounded initial condition $\lambda(0)$, given that $\underline{a}(i) = \underline{a}^*$, $\rho(i) = \rho^*$, and $\underline{b}(i) = \underline{b}^*$ $\forall i$, is

$$0 < \mu < \mu_\lambda \equiv 2MZ^{-2}$$

where $Z = \max_k\{|\alpha(k) - \beta(k)|\}$.

Proof: See the Appendix. □

The mixing parameter λ will play an important role when Gaussian and/or non-Gaussian signals are present in the system. Depending on which one is more representative, the mixing parameter should be adjusted accordingly.

The validity of the above theoretical restrictions on the step sizes for convergence will also be experimentally demonstrated in the applications.

V. Applications in Image Processing

In this section, we verify the proper operation of the training process (8) and illustrate the use of MRL-filters in problems of image processing such as system identification and image restoration.

As proposed in [13], the images are scanned twice during the training process, following a zig zag path from top to bottom, and then from bottom to top. To define the local input vector $\underline{x}$ at each pixel, a square window centered around it with size $n = (2m+1)^2$, $m = 1, 2, \cdots$, is defined such that $\underline{x}$ is the corresponding square matrix transformed to a vector via column-by-column indexing. The vectors $\underline{a}$ and $\underline{b}$ should be interpreted the same way.

Following our design methodology, a plot of an error measurement versus iteration number is provided. This plot is usually called a learning curve. Moreover, for numerical robustness, the unit sample function $q(v)$ is approximated by $q_\sigma(v) = \exp[-\frac{1}{2}(v/\sigma)^2]$, with $\sigma = 0.001$. The SNR's reported in all experiments are the PSNR's, equal to

$$\text{PSNR} = 20\,\log_{10}\left(\frac{\text{Signal peak-to-peak range}}{\sqrt{\text{Var(error signal)}}}\right).$$

We normalized the image pixel values to be in the range [0, 1].

A. Experiment 1: System Identification

The goal here is to experimentally demonstrate convergence or divergence behaviors of the training process (8), using the assumptions of Section IV. Thus, we explore the system identification problem of tracking one element of $\underline{w}^*$, given that we know the original and transformed images, and the other three elements of $\underline{w}^*$. In this way, the input signal is the original image, and the desired signal is the image transformed by the corresponding MRL-filter defined via $\underline{w}^*$. We considered both cases of noiseless and noisy system identification, with $(M = 10)$ and without $(M = 1)$ averaging of the error sequence in the cost function J. The noisy case was generated by adding a uniformly distributed noise with a PSNR $= 45$ dB. The plotted learning error was the percent normalized max absolute error

$$\xi(i) \equiv \frac{\|\underline{w}(i) - \underline{w}^*\|_\infty}{\|\underline{w}^*\|_\infty} \times 100.$$

Our criterion for stopping the iterations was $\xi(i) \leq \xi_c$, where the threshold ξ_c depended on the parameter type and the

TABLE I
RESULTS OF EXPERIMENT 1. THEORETICAL UPPER BOUNDS FOR THE STEP SIZE μ (PROPOSITIONS 4–7) WITH THE CORRESPONDING CASES OF CONVERGENCE OR DIVERGENCE

Experiment 1 (Figures 4–5)			
$\underline{a}^* = \underline{b}^* = \frac{1}{4}\begin{bmatrix}0&1&0\\1&0&1\\0&1&0\end{bmatrix}$		$r^* = 8$ $\lambda^* = \frac{1}{4}$	
$\mu_a = 32M$, $\mu_\rho = 1.61$			
$\mu_b = 0.40M$, $\mu_\lambda = 4.59M$			
$\underline{a}$ ($\underline{a}_0 = \underline{0}$)			
Noiseless ($\xi_c = 10^{-4}\%$, M = 1)		Noisy ($\xi_c = 10\%$, M = 10)	
μ	#Iterations	μ	#Iterations
20	3168	20	3152
40	Divergence	40	1888
ρ ($\rho_0 = 0 \Rightarrow r_0 = 5$)			
Noiseless ($\xi_c = 10^{-4}\%$, M = 1)		Noisy ($\xi_c = 10^{-4}\%$, M = 10)	
μ	#Iterations	μ	#Iterations
60	10	60	10
120	4	120	5
$\underline{b}$ ($\underline{b}_0 = \underline{0}$)			
Noiseless ($\xi_c = 10^{-4}\%$, M = 1)		Noisy ($\xi_c = 1\%$, M = 10)	
μ	#Iterations	μ	#Iterations
0.4	52,194	0.4	19,063
0.8	Divergence	0.8	10,836
λ ($\lambda_0 = 0.5$)			
Noiseless ($\xi_c = 10^{-4}\%$, M = 1)		Noisy ($\xi_c = 1\%$, M = 10)	
μ	#Iterations	μ	#Iterations
100	66	100	13
1000	Divergence	1000	18

noise level. For the case of tracking the optimal rank r^*, we actually plotted $\xi(i)$ using the integer rank r instead of the real parameter ρ.

Table I contains the parameters of the MRL-filter used for system identification, the corresponding theoretical upper bounds for the step size μ, and parts of the simulation results. The training images and the resulting learning curves are shown in Figs. 4 and 5. As a main conclusion from this experiment, whose results are typical of many other similar experiments we have performed, in a supervised system identification task our design algorithm converges fast to the real parameters of the MRL-filter within small error distances. Observe also that the learning curves of the rank parameter r presented almost no dependence on small additive perturbations, so that the same threshold ξ_c could be employed in both noiseless and noisy cases. In contrast, larger thresholds ξ_c had to be used with the other filter parameters in the noisy cases; as expected, the additive filter parameter $\underline{a}$ was the most sensitive to additive perturbations (largest ξ_c). Furthermore, the rates of convergence for the parameters $\underline{a}$, r, and λ were in general much faster than for the parameter $\underline{b}$. Hence, it is faster to design the morphological/rank part than the linear part of the MRL-filter using the LMS approach.

B. Experiment 2: Image Restoration

The goal here it to restore an image corrupted by non-Gaussian noise. Hence, the input signal is a noisy image, and the desired signal is the original (noiseless) image. The noisy image for training the filter was generated by first corrupting the original image with a 47 dB additive Gaussian white noise, and then with a 10% multivalued impulse noise. After the MRL-filter is designed, another noisy image (with similar type of perturbation) is used for testing. The optimal filter parameters were estimated after scanning the image twice during the training process. We used the training algorithm (8) with $M = 1$ and $\mu = 0.1$, and started the process with an unbiased combination between a flat median and the identity, i.e.,

$$\underline{a}_0 = \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix},\ \underline{b}_0 = \begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix},\ \rho_0 = 0,\ \lambda_0 = 0.5.$$

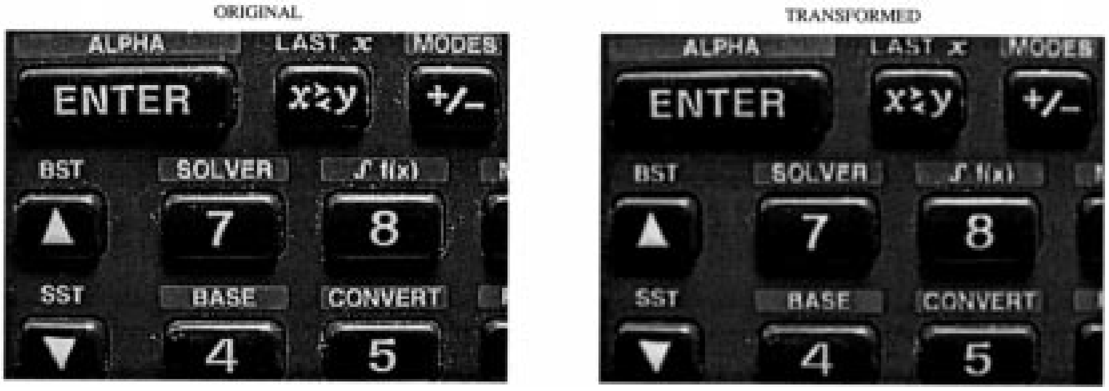
The final trained parameters of the filter were

$$\underline{a} = \begin{bmatrix}0.75&0.00&0.05\\-0.46&-0.01&0.71\\-0.09&-0.02&-0.51\end{bmatrix},$$
$$\underline{b} = \begin{bmatrix}0.01&0.19&-0.01\\0.13&0.86&0.07\\0.00&0.13&-0.02\end{bmatrix},\ r = 5,\ \lambda = 0.98$$

which represents a biased combination between a nonflat median filter and a linear FIR filter, where some elements of $\underline{a}$ and $\underline{b}$ present more influence in the filtering process.

Fig. 6 shows the results of using the designed MRL-filter with a test image, and its comparison with a flat median filter of the same window size. The noisy image used for training is not included there because the (noisy) images used for training and testing are simply different realizations of the same perturbation process. Observe that the MRL-filter outperformed the median filter by about 3 dB. Spatial error plots are also included, where lighter areas indicate higher errors, and they clearly show that the MRL-filter better preserves the image structure.

For the type of noise used in this experiment, we must have at least part of the original (noiseless) image, otherwise we



(a) (b)

Fig. 4. Images of Experiment 1. (a) Original image (211 × 301). (b) Image (a) transformed by the MRL-filter indicated in Table I.

would not be able to provide a good estimate to the optimal filter parameters during the training process (8). In order to validate this point, we repeated Experiment 2 using 100 × 100 subimages of the training image (only 17% of the pixels), and the resulting MRL-filter still outperformed the median filter by about 2.3 dB (that is, 0.5 dB smaller than the result obtained when using full images). There are situations, however, where we can use only the noisy image together with some filter constraints and design the filter that is closest to the identity [13], [14]. But this approach is only appropriate for certain types of impulse noise.

An exhaustive comparison of different filter structures in problems of system identification and noise cancellation is beyond the scope of this paper. Nevertheless, Experiment 2 was extended with the adaptive design of a 3 × 3 L-filter [1] under the same conditions. Starting the L-filter with a flat median, even after scanning the image four times during the training process, the resulting L-filter was just 0.2 dB better than the (flat) median filter.

VI. Conclusions

A new hybrid nonlinear and linear filter was introduced in this paper and termed the *MRL-filter*. It consists of a linear combination between a morphological/rank filter and a linear filter. The main motivation for its definition was the need to have a system capable of dealing with combinations of Gaussian and non-Gaussian noises in signals but also presenting characteristics of a morphological system in some cases.

The nonlinear component of the MRL-filter is defined by a morphological structuring element (vector of additive weight coefficients) and a rank parameter. The linear component is defined by a vector of multiplicative weight coefficients. A mixing parameter is then employed to control the contribution of each component.

An adaptive scheme was then proposed to design this filter, and some new ideas were developed to overcome the problem of nondifferentiability of rank functions. Together with some design choices, the resulting adaptation algorithm was very simple and based on the averaged LMS algorithm. Furthermore, each subset of weight coefficients could be designed individually.

Convergence issues were discussed, and theoretical upper bounds μ_w for the step size μ were proposed, such that the corresponding adaptation algorithms could lead to convergent behavior if $0 < \mu < \mu_w$. An interesting result was that the step bound μ_w for designing the weights for the morphological/rank part does not depend on the input data. Mathematical proofs were provided for the convergence conditions.

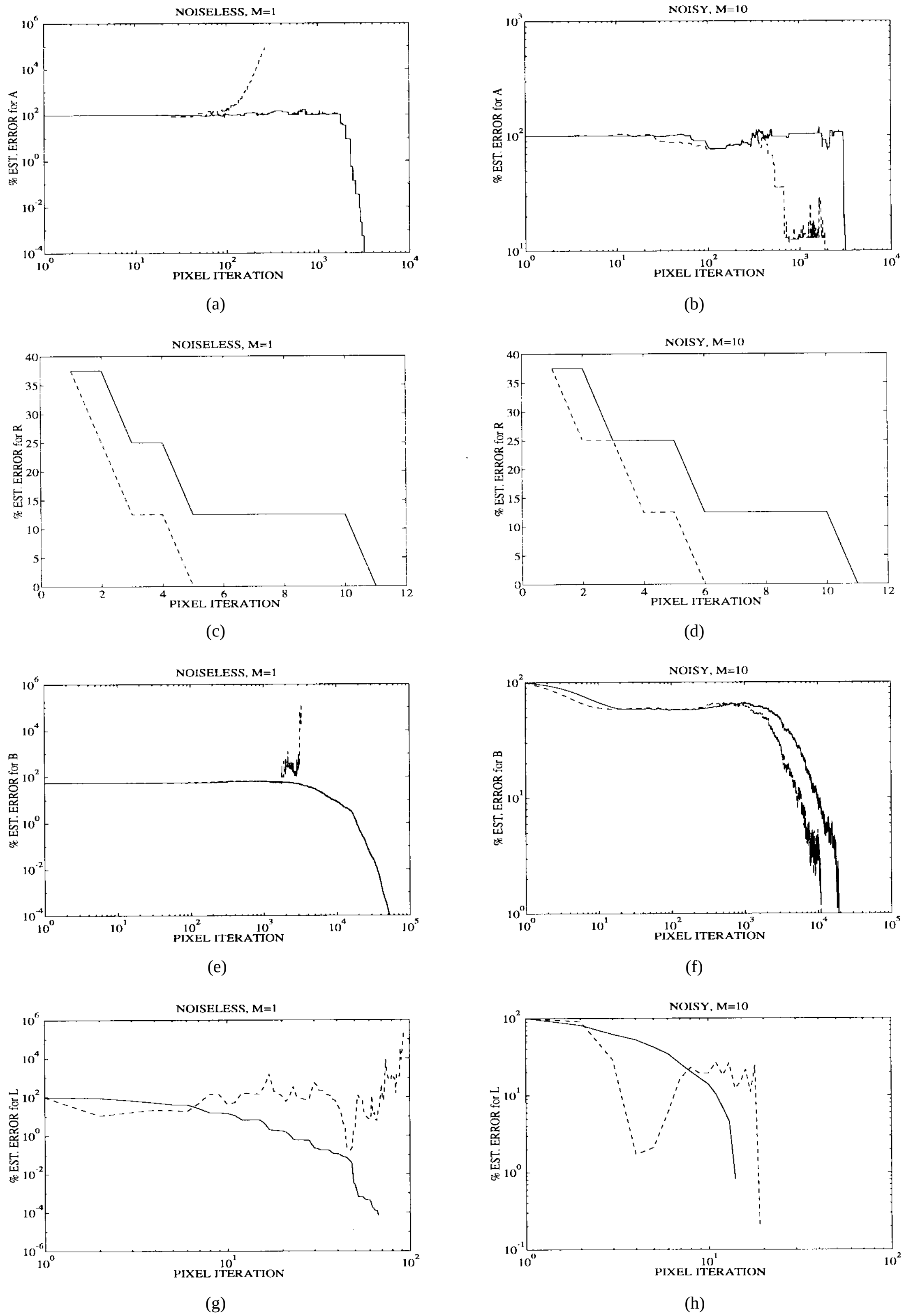
Finally, some applications to system identification and image restoration were illustrated. Our results agreed with the above stated theoretical developments and pointed out the fact that designing morphological/rank filters is more efficient than designing linear FIR filters via the LMS approach, in the sense of faster convergence to the desired filter coefficients. Furthermore, the MRL-filters could outperform the standard flat median filter (in terms of SNR's) in restoring noisy signals. Our results using MRL-filters are encouraging and demonstrate the effectiveness of these filters for certain image/signal processing tasks.

Comparing our design methodology for the nonlinear filter component with the approach in [13] and [14], some extensions and improvements can be pointed out: Theoretical proofs of convergence are provided; no Dirac delta function is used; the notion of "rank indicator vector" and "smoothed impulses" are introduced. Our approach is simpler, more intuitive, and numerically more robust.

Although we defined MRL-filters as a shift-invariant system, we can directly use the proposed design procedure to extend the idea for a shift-variant (adaptive) system. This approach can be useful when the image/signal presents strong nonstationary spatial/temporal characteristics. We are currently investigating the application of MRL-filters as processing nodes in a general class of multilayer networks used for pattern recognition [11].

Appendix A
Proof of Proposition 1

a) Direct consequence of Definition 4.



(a) (b) (c) (d) (e) (f) (g) (h)

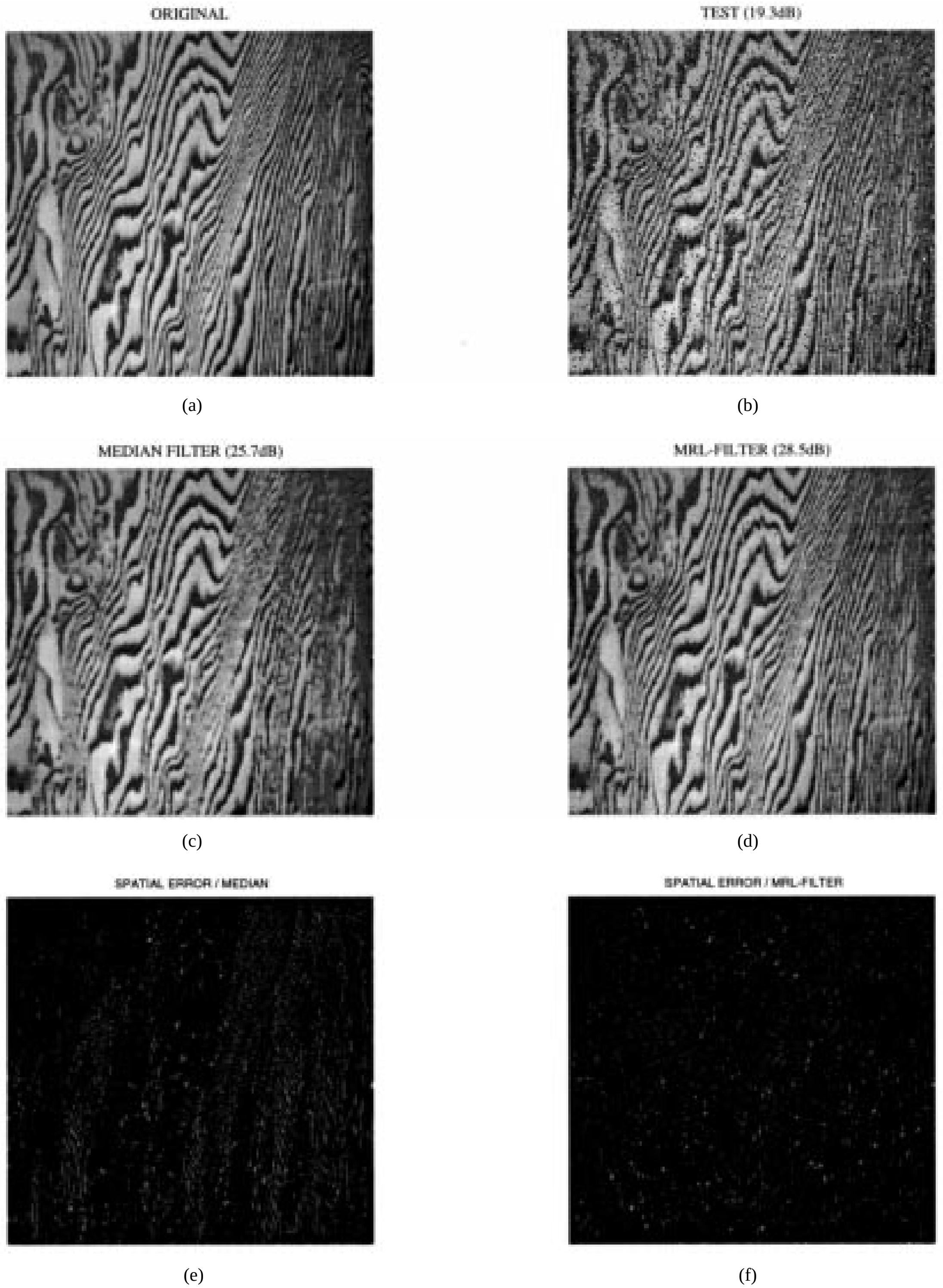
Fig. 5. Learning curves of Experiment 1. (a)–(h) Learning curves of the filter parameters (from top to bottom), for noiseless ($M = 1$) and noisy ($M = 10$) signals (from left to right). The solid and dashed lines correspond, respectively, to the small and large values of the step size μ.

b) If $\underline{c} = (c_1, c_2, \cdots, c_n)$, by Definition 4

$$c_m = \frac{q(z - t_m)}{\sum_{j=1}^{n} q(z - t_j)}, \qquad m = 1, 2, \cdots, n.$$

From Definition 3 it follows that

$$c_m = \begin{cases} 1/N, & \text{if } z = t_m \\ 0, & \text{otherwise} \end{cases}$$



(a) (b) (c) (d) (e) (f)

Fig. 6. Experiment 2. (a) Original texture image (240 × 250). (b) Test image: image (a) corrupted by a hybrid 47 dB additive Gaussian white noise and 10% multivalued impulse noise (PSNR = 19.3 dB). (c) Image (b) restored by a flat 3 × 3 median filter (PSNR = 25.7 dB). (d) Image (b) restored by the designed 3 × 3 MRL-filter (PSNR = 28.5 dB). (e) Spatial error map of the flat median filter. (f) Spatial error map of the MRL-filter.

where N is the number of inputs t_j that are equal to the output $z = \mathcal{R}_r(\underline{t})$. Therefore

$$\underline{c} \cdot \underline{t}' = \sum_{m=1}^{n} t_m c_m = \frac{1}{N} \underbrace{(z + z + \cdots + z)}_{N \text{ times}} = \mathcal{R}_r(\underline{t}).$$

c) Direct consequence of a) and b).

d) For a fixed rank r, the number N of inputs that are equal to the output varies from 1 to n. Hence, the total number of all possible different values of $\underline{c}$ in $\mathbb{R}^n$ is

$$\sum_{N=1}^{n} \frac{n!}{N!(n-N)!} = 2^n - 1.$$

Now consider $\underline{t}_0 = (t_{0,1}, t_{0,2}, \cdots, t_{0,n})$ in $\mathbb{R}^n$ such that $t_{0,i} \neq t_{0,j}\ \forall i \neq j$, and any $\underline{t} \in \mathbb{R}^n$ such that $\|\underline{t} - \underline{t}_0\|_\infty < \frac{1}{2}\min_{i \neq j}|t_{0,i} - t_{0,j}|$. Then, $\underline{t}$ and $\underline{t}_0$ have the same ordering, and hence $\underline{c}(\underline{t}, r) = \underline{c}(\underline{t}_0, r)$. □

Appendix B
Proof of Proposition 2

From Proposition 1b and d, if $\underline{c}(\underline{t}, r)$ is constant in a neighborhood of $\underline{t}_0$, then $\mathcal{R}_r(\underline{t}) = \underline{c}(\underline{t}_0, r) \cdot \underline{t}'\ \forall \underline{t}$ in that region, so that

$$\left.\frac{\partial \mathcal{R}_r(\underline{t})}{\partial \underline{t}}\right|_{\underline{t}=\underline{t}_0} = \left.\frac{\partial}{\partial \underline{t}}\left[\underline{c}(\underline{t}_0, r) \cdot \underline{t}'\right]\right|_{\underline{t}=\underline{t}_0} = \underline{c}(\underline{t}_0, r).$$

Otherwise, $\mathcal{R}_r(\underline{t})$ is not differentiable. □

Appendix C
Proof of Proposition 3

a) Direct consequence of Definition 5.

b) Since $q_\sigma(v) \to q(v)$ as $\sigma \to 0$, then from Definition 5 it follows directly that $\underline{c}_\sigma \to \underline{c}$ as $\sigma \to 0$.

c) Direct consequence of b), Definition 5, and Proposition 1b.

d) Since $q_\sigma(v) \to 1$ as $\sigma \to \infty$, then from Definition 5 it follows directly that

$$\underline{c}_\sigma \to \frac{1}{\underline{1} \cdot \underline{1}'}\underline{1} = \frac{1}{n}\underline{1} \text{ as } \sigma \to \infty.$$

e) Direct consequence of d and Definition 5. □

Appendix D
Proof of Proposition 4

Assume convergence in the training of $\underline{a}$. Since ρ, $\underline{b}$, and λ are optimally fixed, $\lambda \neq 0$, then from (8)–(10) the training process of the jth element of $\underline{a}$ is

$$a_j(i+1) = a_j(i) + \frac{\lambda\mu}{M}\sum_{k=i-M+1}^{i} e(k)c_j(k), \quad i = 0, 1, 2, \cdots \tag{D.1}$$

where $c_j(k)$ is the jth element of $\underline{c}[\underline{x}(k) + \underline{a}(i), r]$. Let $\underline{a}^*$ be the desired vector, and consider the subsequences $\{\underline{x}[m_1(i)]\}$ and $\{\underline{x}[m_2(i)]\}$ of training samples such that

$$\mathcal{R}_r\{\underline{x}[m_1(i)] + \underline{a}^*\} = x_j[m_1(i)] + a_j^* \tag{D.2}$$

and

$$\mathcal{R}_r\{\underline{x}[m_2(i)] + \underline{a}[m_2(i)]\} = x_j[m_2(i)] + a_j[m_2(i)]. \tag{D.3}$$

The iteration (D.1) is convergent if $e(i) \to 0$ as $i \to \infty$. For a convergent process, we must have $\{m(i)\colon m(i) = m_1(i) = m_2(i)\} \neq \emptyset$, because otherwise we contradict our assumption of convergence. Using the training samples $\{\underline{x}[m(i)]\}$, then from (1), (D.2), and (D.3)

$$e[m(i)] = d[m(i)] - y[m(i)] = \lambda\{a_j^* - a_j[m(i)]\} \tag{D.4}$$

so that $|a_j[m(i)] - a_j^*| \to 0$ if $e[m(i)] \to 0$. By substituting (D.4) in (D.1), subtracting a_j^* from both sides of the resulting equation and taking the absolute value, it follows that for $i = 0, 1, 2, \cdots$

$$\begin{aligned} 0 &\leq \left|1 - \lambda^2\frac{\mu}{M}c_j[m(i)]\right|\varepsilon_j[m(i)] \leq \varepsilon_j[m(i+1)] \\ &\leq \left|1 - \lambda^2\frac{\mu}{M}c_j[m(i)]\right|\varepsilon_j[m(i)] + A\tilde{e}[m(i)] \end{aligned} \tag{D.5}$$

where

$$\begin{aligned} \varepsilon_j[m(i)] &= |a_j[m(i)] - a_j^*| \\ A &= |\lambda|\mu(M-1)/M \\ \tilde{e}[m(i)] &= \max_{i-M+1 \leq k \leq i-1}\{|e[m(k)]|\}. \end{aligned}$$

But for a convergent process $\tilde{e}[m(i)] \to 0$ as $i \to \infty$, so that from (D.5) a necessary condition for convergence is $|1 - \lambda^2(\mu/M)c_j[m(i)]| < 1\ \forall m(i)$. Furthermore, from (D.3) $1/n \leq c_j[m(i)] \leq 1$, so that the least upper bound μ_a for the step size μ is obtained if

$$0 < \mu < \mu_a = 2M\lambda^{-2}. \tag{D.6}$$

The same result can be similarly derived for every element a_j of $\underline{a}$. Therefore the restriction (D.6) is a necessary condition for $\|\underline{a}(i) - \underline{a}^*\| \to 0$ as $i \to \infty$ for any bounded initial condition $\underline{a}(0)$. □

Appendix E
Proof of Proposition 5

Since $\underline{a}$, $\underline{b}$, and λ are optimally fixed, $\lambda \neq 0$, then from (8), (9), and (11) the training process of the parameter ρ is

$$\rho(i+1) = \rho(i) + \frac{\lambda\mu}{M}\sum_{k=i-M+1}^{i} e(k)s(k), \quad i = 0, 1, 2, \cdots. \tag{E.1}$$

From (3), the corresponding training process of the rank parameter r is

$$r(i+1) = \lfloor r'(i+1) + 0.5 \rfloor, \qquad i = 0, 1, 2, \cdots \tag{E.2}$$

where

$$r' = n - \frac{n-1}{1 + \exp(-\rho)}. \tag{E.3}$$

Whenever $r(i) = r^*$, the desired rank, then the updating process stops $[e(k) = 0\ \forall k]$, so that $|\rho(i+1) - \rho(i)| = 0$, thereafter. If $r(0) = r^*$, nothing needs to be done. Otherwise, either $r(0) > r^*$ or $r(0) < r^*$. Since the sequence $\mathcal{S} = [r(i)]$ is bounded, a sufficient condition for $|r(i) - r^*| \to 0$ and $|\rho(i+1) - \rho(i)| \to 0$ as $i \to \infty$ for any bounded initial condition $\rho(0)$ is that $\mathcal{S}$ is either monotone decreasing or monotone increasing. But this condition is guaranteed if

$$|r'(i+1) - r'(i)| < 1 \qquad \forall i. \tag{E.4}$$

From (E.1) and (E.3) we can write

$$r'(i+1) = \frac{[nB(i) - 1]r'(i) - [B(i) - 1]n}{[B(i) - 1]r'(i) + [n - B(i)]} \tag{E.5}$$

where

$$B(i) = \exp\left[-\frac{\lambda\mu}{M}\, B'(i)\right] \tag{E.6}$$

and

$$B'(i) = \sum_{k=i-M+1}^{i} e(k)s(k). \tag{E.7}$$

Since $1 \le r'(i) \le n$, then $[B(i)-1]r'(i)+[n-B(i)] \ge n-1$, such that subtracting $r'(i)$ from (E.5) and taking the absolute value, results

$$|r'(i+1)-r'(i)| \le n|B(i)-1|\, \frac{r'(i)-1}{n-1} \le n|B(i)-1|. \tag{E.8}$$

Hence, from (E.4), (E.6), and (E.8), a sufficient condition for convergence is

$$0 < \mu < \frac{M}{|\lambda|\,|B'(i)|}\, \ln\left(1+\frac{1}{n}\right) < \frac{M}{n|\lambda|\,|B'(i)|}. \tag{E.9}$$

On the other hand, from (11) and (E.7),

$$|B'(i)| \le \frac{n-1}{n} \sum_{k=i-M+1}^{i} |e(k)|. \tag{E.10}$$

Besides this, from (1) and (7)

$$\sum_{k=i-M+1}^{i} |e(k)| \le M \max_k \{|e(k)|\} \le M|\lambda|D \tag{E.11}$$

where $D = \max_{k,l}\{|x(k) - x(l)|\} + \max_{k,l}\{|a_k - a_l|\}$. Substituting (E.11) in (E.10), then from (E.9) the least upper bound μ_ρ for the step size μ is obtained if

$$0 < \mu < \mu_\rho = \frac{\lambda^{-2}}{(n-1)D}$$

such that convergence is guaranteed. □

Appendix F
Proof of Proposition 6

Since $\underline{a}$, ρ, and λ are optimally fixed, $\lambda \neq 1$, then from (8) and (9) the training process of the vector $\underline{b}$ is

$$\begin{aligned}\underline{b}(i+1) = {} & \underline{b}(i) + \frac{(1-\lambda)\mu}{M} \sum_{k=i-M+1}^{i} e(k)\underline{x}(k),\\ & i = 0,\, 1,\, 2,\, \cdots.\end{aligned} \tag{F.1}$$

If this iteration is convergent, then $e(i) \to 0$ as $i \to \infty$. Choose any sample $\underline{x}(i)$, and let $\underline{b}^*$ be the desired vector $\underline{b}$. From (1)

$$e(i) = d(i) - y(i) = (1-\lambda)[\underline{b}^* - \underline{b}(i)] \cdot \underline{x}'(i) \tag{F.2}$$

so that $\|\underline{b}(i) - \underline{b}^*\| \to 0$ if $e(i) \to 0$, given that $\underline{x}(i) \neq \underline{0}$. For the same sample, computing the error at iteration $i+1$, using (F.1) and taking the absolute value, results that for $i = 0,\, 1,\, 2,\, \cdots$

$$\begin{aligned}0 \le & \left|1 - [L(1-\lambda)]^2 n\, \frac{\mu}{M}\right| |e(i)| \le |e(i+1)|\\ \le & \left|1 - [L(1-\lambda)]^2 n\, \frac{\mu}{M}\right| |e(i)| + Cq(i)\end{aligned} \tag{F.3}$$

where

$$\begin{aligned}L &= \max_k \{|x(k)|\}\\ C &= (1-\lambda)^2 \mu\, \frac{M-1}{M}\\ q(i) &= \|\underline{x}(i)\| \max_{i-M+1 \le k \le i-1} \{|e(k)|\, \|\underline{x}(k)\|\}.\end{aligned}$$

From (F.3), a necessary condition for convergence is

$$\left|1 - [L(1-\lambda)]^2 n\, \frac{\mu}{M}\right| < 1$$

or equivalently

$$0 < \mu < \mu_a = \frac{2M}{n}\, [L(1-\lambda)]^{-2}. \tag{F.4}$$

□

Appendix G
Proof of Proposition 7

Since $\underline{a}$, ρ, and $\underline{b}$ are optimally fixed, then from (8) and (9) the training process of the parameter λ is

$$\begin{aligned}\lambda(i+1) = {} & \lambda(i) + \frac{\mu}{M} \sum_{k=i-M+1}^{i} e(k)[\alpha(k) - \beta(k)],\\ & i = 0,\, 1,\, 2,\, \cdots.\end{aligned} \tag{G.1}$$

If this iteration is convergent, then $e(i) \to 0$ as $i \to \infty$. Choose any sample $\underline{x}(i)$, and let λ^* be the desired mixing parameter λ. From (1)

$$e(i) = d(i) - y(i) = [\lambda^* - \lambda(i)][\alpha(i) - \beta(i)] \tag{G.2}$$

so that $|\lambda(i) - \lambda^*| \to 0$ if $e(i) \to 0$, given that $\alpha(i) \neq \beta(i)$. For the same sample, computing the error at iteration $i+1$, using (G.1) and taking the absolute value, results that for $i = 0,\, 1,\, 2,\, \cdots$

$$\begin{aligned}0 \le & \left|1 - \frac{\mu}{M}\, Z^2\right| |e(i)| \le |e(i+1)|\\ \le & \left|1 - \frac{\mu}{M}\, Z^2\right| |e(i)| + D\tilde{z}(i)\end{aligned} \tag{G.3}$$

where

$$\begin{aligned}Z &= \max_k \{|\alpha(k) - \beta(k)|\}\\ D &= \mu\, \frac{M-1}{M}\end{aligned}$$

$$\tilde{z}(i) = |\alpha(i) - \beta(i)| \max_{i-M+1 \le k \le i-1} \{|e_k|\, |\alpha(k) - \beta(k)|\}.$$

From (G.3), a necessary condition for convergence is

$$\left|1 - \frac{\mu}{M}\, Z^2\right| < 1$$

or equivalently

$$0 < \mu < \mu_\lambda = 2MZ^{-2}. \tag{G.4}$$

□

ACKNOWLEDGMENT

The authors thank D. Mumford for his suggestion to smooth the unit sample function with Gaussians, D. A. F. Florêncio for his suggestion about smoothed rank functions and discussions during the beginning of this work, and all the reviewers, for their helpful comments.

REFERENCES

[1] A. C. Bovik, T. S. Huang, and D. C. Munson, "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-31, pp. 1342–1350, Dec. 1983.

[2] P. M. Clarkson, *Optimal and Adaptive Signal Processing.* Boca Raton, FL: CRC, 1993.

[3] E. J. Coyle and J. H. Lin, "Stack filters and the mean absolute error criterion," *IEEE Trans. Acoust. Speech Signal Processing,* vol. 36, pp. 1244–1254, Aug. 1988.

[4] P. Heinonen and Y. Neuvo, "FIR-median hybrid filters," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-35, pp. 832–838, June 1987.

[5] C. Kotropoulos and I. Pitas, "Constrained adaptive LMS L-filters," *Signal Process.,* vol. 26, pp. 335–358, 1992.

[6] R. P. Loce and E. R. Dougherty, "Facilitation of optimal binary morphological filter design via structuring element libraries and design constraints," *Opt. Eng.,* vol. 31, pp. 1008–1025, May 1992.

[7] P. Maragos, "A representation theory for morphological image and signal processing," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 11, pp. 586–599, June 1989.

[8] P. Maragos and R. W. Schafer, "Morphological Filters—Part II: Their relations to median, order-statistic, and stack filters," *IEEE Trans. Acoust. Speech, Signal Process.,* vol. ASSP-35, pp. 1170–1184, Aug. 1987; see also vol. ASSP-37, p. 597, Apr. 1989.

[9] F. Palmieri and C. G. Boncelet, "L*l*-filters—A new class of order statistic filters," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 37, pp. 691–701, May 1989.

[10] F. Palmieri, "Adaptive recursive order statistic filters," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing,* 1990, pp. 1229–1232.

[11] L. F. C. Pessoa and P. Maragos, "Neural networks with hybrid morphological/rank/linear nodes and their application to handwritten character recognition," in *Proc. 9th Europ. Signal Processing Conf.,* Sept. 1998.

[12] I. Pitas and A. N. Venetsanopoulos, "Adaptive filters based on order statistics," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. 39, pp. 518–522, Feb. 1991.

[13] P. Salembier, "Adaptive rank order based filters," *Signal Process.,* vol. 27, pp. 1–25, Apr. 1992.

[14] ——, "Structuring element adaptation for morphological filters," *J. Vis. Commun. Image Represent.,* vol. 3, pp. 115–136, June 1992.

[15] J. Serra, *Image Analysis and Mathematical Morphology.* New York: Academic, 1982, vol. 1.

[16] S. S. Wilson, "Morphological networks," in *Proc. SPIE Visual Communication and Image Processing IV,* 1989, vol. 1199, pp. 483–493.

[17] P. Yang and P. Maragos, "Min-max classifiers: Learnability, design, and application," *Pattern Recognit.,* vol. 28, pp. 879–899, 1995.

Lúcio F. C. Pessoa received the B.S.E.E. and M.S.E.E. degrees from the Federal University of Pernambuco (UFPE), Recife, Brazil, in 1990 and 1992, respectively, and the Ph.D. degree from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 1997.

From 1995 to 1997, he was a Graduate Research Assistant in the Center for Signal and Image Processing (formerly known as DSL Laboratory) of the School of Electrical and Computer Engineering, Georgia Tech. In March 1998, he joined the Broadband Division of Motorola Semiconductor Products Sector, Austin, TX. His Ph.D. research project was in the design of optimal morphological systems with applications to image processing and neural networks. His main research interests include image processing, computer vision, pattern recognition, and communications.

Dr. Pessoa was honored with several research fellowships during his studies at UFPE, and in 1990 he received the Brazilian IEEE/CEMIG Award for Technology.

Petros Maragos (S'81–M'85–SM'91–F'95) received the Diploma degree in electrical engineering from the National Technical University of Athens, Greece, in 1980, and the M.S.E.E. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 1982 and 1985.

In 1985, he joined the faculty of the Division of Applied Sciences, Harvard University, Cambridge, MA, where he was an Assistant (1985–1989) and Associate Professor (1989–1993) of Electrical Engineering, affiliated with the interdisciplinary Harvard Robotics Laboratory. He has also been a consultant to several industry research groups, including Xerox's research on document image analysis. In June 1993, he joined the faculty of the Georgia Institute of Technology, where he is currently an Associate Professor of Electrical and Computer Engineering. During parts of 1995 to 1998, he also worked as a Visiting Professor at the National Technical University of Athens, Greece, and as a Senior Researcher at the Greek Institute for Language and Speech Processing.

Dr. Maragos has received several awards, including a 1987 U.S. National Science Foundation Presidential Young Investigator Award; the 1988 IEEE Signal Processing Society's Paper Award for the paper, "Morphological filters"; the 1994 IEEE Signal Processing Society's Senior Award and the 1995 IEEE Baker Award for the paper "Energy separation in signal modulations with application to speech analysis" (co-recipient); and the 1996 Pattern Recognition Society's Honorable Mention Award for the paper "Min–Max Classifiers" (co-recipient). He has also served as Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and Guest Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING; General Chairman for the 1992 SPIE Conference on Visual Communications and Image Processing, and Co-chairman for the 1996 International Symposium on Mathematical Morphology. He has been a member of two IEEE DSP Committees, and President of the International Society for Mathematical Morphology.