P. Maragos and L. F.C. Pessoa, "Morphological Filtering for Image Enhancement and Detection",
in *Image and Video Processing Handbook (First Edition)*, edited by A. C. Bovik, Acad. Press, 2000, pp.101-116.

MORPHOLOGICAL FILTERING FOR IMAGE ENHANCEMENT AND DETECTION

Petros Maragos[1] and Lúcio F. C. Pessoa[2]

[1] Dept. of Electrical & Computer Engineering, National Technical University of Athens, Zografou 15773, Athens, Greece.

[2] Motorola, Inc., 3501 Ed Bluestein Blvd., Austin, TX 78721, U.S.A.

Email: Petros.Maragos@cs.ntua.gr, Lucio_Pessoa@email.mot.com

March 1, 1999

1 Introduction

The goals of image enhancement include the improvement of the visibility and perceptibility of the various regions into which an image can be partitioned and of the detectability of the image features inside these regions. These goals include tasks such as: cleaning the image from various types of noise; enhancing the contrast among adjacent regions or features; simplifying the image via selective smoothing or elimination of features at certain scales and retaining only features at certain desirable scales. While traditional approaches for solving the above tasks have used mainly tools of linear systems, there is a growing understanding that linear approaches are not well suitable or even fail to solve problems involving geometrical aspects of the image. Thus there is a need for nonlinear approaches. A powerful nonlinear methodology that can successfully solve the above problems is mathematical morphology.

Mathematical morphology is a set- and lattice-theoretic methodology for image analysis, which aims at quantitatively describing the geometrical structure of image objects. It was initiated [17] in the late 1960's to analyze binary images from geological and biomedical data as well as to formalize and extend earlier or parallel work [13, 12] on binary pattern recognition based on cellular automata and Boolean/threshold logic. In the late 1970's it was extended to gray-level images [17]. In the mid 1980's it was brought to the mainstream of image/signal processing and related to other nonlinear filtering approaches [7, 8]. Finally, in the late 1980's and 1990's it was generalized to arbitrary lattices [18, 2]. The above evolution of ideas has formed what we call nowadays the field of **morphological image processing**, which is a broad and coherent collection of theoretical concepts, nonlinear filters, design methodologies, and applications systems. Its rich theoretical framework, algorithmic efficiency, easy implementability on special hardware, and suitability for many shape-oriented problems have propelled its widespread usage and further advancement by many academic and industry groups working on various problems in image processing, computer vision, and pattern recognition.

This chapter provides a brief introduction to the application of morphological image processing to image enhancement and detection. There are several motivations for using morphological filters for such problems. First, it is of paramount importance to preserve, uncover, or detect the geometric structure of image objects. Thus, morphological filters which are more suitable than linear filters for shape analysis, play a major role for geometry-based enhancement and detection. Further, they offer efficient solutions to other nonlinear tasks such as non-Gaussian noise suppression. This task can also be accomplished (with similar performance) by a closely related class of nonlinear systems, the median, rank and stack filters, which also outperform linear filters in non-Gaussian noise suppression. Finally, the elementary morphological operators[1] are the building blocks for large classes of nonlinear image processing systems, which include rank and stack filters.

2 Morphological Image Operators

2.1 Morphological Filters for Binary Images

Given a sampled[2] binary image signal $f[x]$ with values 1 for the image object and 0 for the background, typical image transformations involving a moving window set $W = \{y_1, y_2, ..., y_n\}$ of n sample indexes would be

$$\psi_b(f)[x] = b(f[x - y_1], ..., f[x - y_n]) \tag{1}$$

where $b(v_1, ..., v_n)$ is a Boolean function of n variables. The mapping $f \mapsto \psi_b(f)$ is called a *Boolean filter*. By varying the Boolean function b, a large variety of Boolean filters can be obtained. For example, choosing a Boolean AND for b would *shrink* the input image object, whereas a Boolean OR would *expand* it. Numerous other Boolean filters are possible, since there are 2^{2^n} possible Boolean functions of n variables. The main applications of such Boolean image operations have been in biomedical image processing, character recognition, object detection, and general 2D shape analysis [13, 12].

Among the important concepts offered by mathematical morphology was to use *sets* to represent binary images and set operations to represent binary image transformations. Specifically, given a binary image, let the object be represented by the set X and its background by the set complement X^c. The Boolean OR transformation of X by a (window) set B is equivalent to the Minkowski set addition $\oplus$, also called **dilation**, of X by B:

$$X \oplus B \equiv \{x + y : x \in X,\ y \in B\} = \bigcup_{y \in B} X_{+y} \tag{2}$$

where $X_{+y} \equiv \{x + y : x \in X\}$ is the *translation* of X along the vector y. Likewise, if $B^r \equiv \{x : -x \in B\}$ is the *reflection* of B with respect to the origin, the Boolean AND transformation of X by B^r is equivalent to the Minkowski set subtraction $\ominus$, also called **erosion**, of X by B:

$$X \ominus B \equiv \{x : B_{+x} \subseteq X\} = \bigcap_{y \in B} X_{-y} \tag{3}$$

[1] The term 'morphological operator', which means a morphological signal transformation, shall be used interchangeably with 'morphological filter', in analogy to the terminology 'rank or linear filter'.

[2] Signals of a continuous variable $x \in \mathbb{R}^d$ are usually denoted by $f(x)$, whereas for signals with discrete variable $x \in \mathbb{Z}^d$ we write $f[x]$. $\mathbb{R}$ and $\mathbb{Z}$ denote, respectively, the set of reals and integers.

Cascading erosion and dilation creates two other operations, the **opening** $X \circ B \equiv (X \ominus B) \oplus B$ and the **closing** $X \bullet B \equiv (X \oplus B) \ominus B$ of X by B. In applications, B is usually called a *structuring element* and has a simple geometrical shape and a size smaller than the image X. If B has a regular shape, e.g., a small disk, then both opening and closing act as nonlinear filters that smooth the contours of the input image. Namely, if X is viewed as a flat island, the opening suppresses the sharp capes and cuts the narrow isthmuses of X, whereas the closing fills in the thin gulfs and small holes.

There is a *duality* between dilation and erosion since $X \oplus B = (X^c \ominus B^r)^c$; i.e., dilation of an image object by B is equivalent to eroding its background by B^r and complementing the result. A similar duality exists between closing and opening.

2.2 Morphological Filters for Graylevel Images

Extending morphological operators from binary to graylevel images can be done by using set representations of signals and transforming these input sets via morphological set operations. Thus, consider an image signal $f(x)$ defined on the continuous or discrete plane $\mathbb{ID} = \mathbb{R}^2$ or $\mathbb{Z}^2$ and assuming values in $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. Thresholding f at all amplitude levels v produces an ensemble of binary images represented by the **threshold sets**

$$\Theta_v(f) \equiv \{x \in \mathbb{ID} : f(x) \geq v\}\ , \quad -\infty < v < +\infty \tag{4}$$

The image can be exactly reconstructed from all its threshold sets since

$$f(x) = \sup\{v \in \mathbb{R} : x \in \Theta_v(f)\} \tag{5}$$

where 'sup' denotes supremum[3]. Transforming each threshold set of the input signal f by a set operator Ψ and viewing the transformed sets as threshold sets of a new image creates [17, 7] a **flat** image operator ψ whose output signal is

$$\psi(f)(x) = \sup\{v \in \mathbb{R} : x \in \Psi[\Theta_v(f)]\} \tag{6}$$

For example, if Ψ is the set dilation and erosion by B, the above procedure creates the two most elementary morphological image operators: the dilation and erosion of $f(x)$ by a set B:

$$(f \oplus B)(x) \quad \equiv \quad \bigvee_{y \in B} f(x - y) \tag{7}$$

$$(f \ominus B)(x) \quad \equiv \quad \bigwedge_{y \in B} f(x + y) \tag{8}$$

where $\bigvee$ denotes supremum (or maximum for finite B) and $\bigwedge$ denotes infimum (or minimum for finite B). Flat erosion (dilation) of a function f by a small convex set B reduces (increases) the peaks (valleys) and enlarges the minima (maxima) of the function. The flat opening $f \circ B = (f \ominus B) \oplus B$ of f by B smooths the graph of f from below by cutting down its peaks, whereas the closing $f \bullet B = (f \oplus B) \ominus B$ smooths it from above by filling up its valleys.

[3]Given a set X of real numbers, the *supremum* of X is its lowest upper bound. If X is finite (or infinite but closed from above), its supremum coincides with its maximum.

The most general translation-invariant morphological dilation and erosion of a graylevel image signal $f(x)$ by another signal g are:

$$(f \oplus g)(x) \equiv \bigvee_{y \in \mathbb{D}} f(x-y) + g(y) \tag{9}$$

$$(f \ominus g)(x) \equiv \bigwedge_{y \in \mathbb{D}} f(x+y) - g(y) \tag{10}$$

Note that signal dilation is a nonlinear convolution where the sum-of-products in the standard linear convolution is replaced by a max-of-sums.

2.3 Universality of Morphological Operators[4]

Dilations or erosions can be combined in many ways to create more complex morphological operators that can solve a broad variety of problems in image analysis and nonlinear filtering. Their versatility is further strengthened by a theory outlined in [7, 8] that represents a broad class of nonlinear and linear operators as a minimal combination of erosions or dilations. Here we summarize the main results of this theory restricting our discussion only to discrete 2D image signals.

Any *translation-invariant* set operator Ψ is uniquely characterized by its **kernel** $\mathrm{Ker}(\Psi) \equiv \{X \in \mathbb{Z}^2 : 0 \in \Psi(X)\}$. The kernel representation requires an infinite number of erosions or dilations. A more efficient (requiring less erosions) representation uses only a substructure of the kernel, its **basis** $\mathrm{Bas}(\Psi)$, defined as the collection of kernel elements that are *minimal* with respect to the partial ordering $\subseteq$. If Ψ is also *increasing* (i.e., $X \subseteq Y \Longrightarrow \Psi(X) \subseteq \Psi(Y)$) and *upper semicontinuous* (i.e., $\Psi(\bigcap_n X_n) = \bigcap_n \Psi(X_n)$ for any decreasing set sequence X_n), then Ψ has a nonempty basis and can be represented exactly as a union of erosions by its basis sets:

$$\Psi(X) = \bigcup_{A \in \mathrm{Bas}(\Psi)} X \ominus A \tag{11}$$

The morphological basis representation has also been extended to graylevel signal operators. As a special case, if ϕ is a flat signal operator as in (6) that is translation-invariant and commutes with thresholding, then ϕ can be represented as a supremum of erosions by the basis sets of its corresponding set operator Φ:

$$\phi(f) = \bigvee_{A \in \mathrm{Bas}(\Phi)} f \ominus A \tag{12}$$

By duality, there is also an alternative representation where a set operator Ψ satisfying the above three assumptions can be realized exactly as the intersection of dilations by the reflected basis sets of its *dual operator* $\Psi^d(X) \equiv [\Psi(X^c)]^c$. There is also a similar dual representation of signal operators as an infimum of dilations.

Given the wide applicability of erosions/dilations, their parallellism, and their simple implementations, the morphological representation theory supports a general purpose image processing (software or hardware) module that can perform erosions/dilations, based on which numerous other complex image operations can be build.

[4]This is a section for mathematically-inclined readers and can be skipped without significant loss of continuity.

2.4 Median, Rank, and Stack Filters

Flat erosion and dilation of a discrete image signal $f[x]$ by a finite window $W = \{y_1, ..., y_n\} \subseteq \mathbb{Z}^2$ is a moving local minimum or maximum. Replacing min/max with a more general rank leads to rank filters. At each location $x \in \mathbb{Z}^2$, sorting the signal values within the reflected and shifted n-point window $(W^r)_{+x}$ in decreasing order and picking the p-th largest value, $p = 1, 2, ..., n$, yields the output signal from the p-th **rank filter**:

$$(f \square_p W)[x] \equiv p\text{-th rank of } (f[x - y_1], ..., f[x - y_n]) \tag{13}$$

For odd n and $p = (n+1)/2$ we obtain the **median** filter. Rank filters and especially medians have been applied mainly to suppress impulse noise or noise whose probability density has heavier tails than the Gaussian for enhancement of image and other signals, since they can remove this type of noise without blurring edges, as would be the case for linear filtering. A discussion of median-type filters can be found in Chapter 3.2.

If the input image is binary, the rank filter output is also binary since sorting preserves a signal's range. Rank filtering of binary images involves only counting of points and no sorting. Namely, if the set $S \subseteq \mathbb{Z}^2$ represents an input binary image, the output set produced by the p-th **rank set filter** is

$$S \square_p W \equiv \{x : \text{card}((W^r)_{+x} \cap S) \geq p\} \tag{14}$$

where $\text{card}(X)$ denotes the cardinality (i.e., number of points) of a set X.

All rank operators *commute with thresholding*; i.e.,

$$\Theta_v[f \square_p W] = [\Theta_v(f)] \square_p W, \quad \forall v \ , \ \forall p. \tag{15}$$

where $\Theta_v(f)$ is the binary image resulting from thresholding f at level v. This property is also shared by all morphological operators that are finite compositions or maxima/minima of flat dilations and erosions by finite structuring elements. All such signal operators ψ that have a corresponding set operator Ψ and commute with thresholding can be alternatively implemented via *threshold superposition* as in (6). Further, since the binary version of all the above discrete translation-invariant finite-window operators can be described by their generating Boolean function as in (1), all that is needed in synthesizing their corresponding graylevel image filters is knowledge of this Boolean function. Specifically, let $f_v[x]$ be the binary images represented by the threshold sets $\Theta_v(f)$ of an input graylevel image $f[x]$. Transforming all f_v with an increasing (i.e., containing no complemented variables) Boolean function $b(u_1, ..., u_n)$ in place of the set operator Ψ in (6) creates a class of nonlinear signal operators via threshold superposition, called **stack filters** [1, 7]

$$\phi_b(f)[x] \equiv \sup\{v \in \mathbb{R} : b(f_v[x - y_1], ..., f_v[x - y_n]) = 1\} \tag{16}$$

The use of Boolean functions facilitates the design of such discrete flat operators with determinable structural properties. Since each increasing Boolean function can be uniquely represented by an irreducible sum (product) of product (sum) terms, and each product (sum) term corresponds to an erosion (dilation), each stack filter can be represented as a finite maximum (minimum) of flat erosions (dilations) [7]. Because of their representation via erosions/dilations (which have a geometric interpretation) and Boolean functions (which are related to mathematical logic), stack

filters can be analyzed or designed not only in terms of their statistical properties for image denoising but also in terms of their geometric and logic properties for preserving selected image structures.

2.5 Morphological Operators and Lattice Theory

A more general formalization [18, 2] of morphological operators views them as operators on complete lattices. A *complete lattice* is a set $\mathcal{L}$ equipped with a partial ordering $\leq$ such that $(\mathcal{L}, \leq)$ has the algebraic structure of a *partially ordered set* where the supremum and infimum of any of its subsets exist in $\mathcal{L}$. For any subset $\mathcal{K} \subseteq \mathcal{L}$, its *supremum* $\bigvee \mathcal{K}$ and *infimum* $\bigwedge \mathcal{K}$ are defined as the lowest (with respect to $\leq$) upper bound and greatest lower bound of $\mathcal{K}$, respectively. The two main examples of complete lattices used in morphological image processing are: (i) the space of all binary images represented by subsets of the plane $\mathbb{ID}$ where the $\bigvee / \bigwedge$ lattice operations are the set union/intersection, and (ii) the space of all graylevel image signals $f : \mathbb{ID} \rightarrow \overline{\mathbb{R}}$ where the $\bigvee / \bigwedge$ lattice operations are the supremum/infimum of sets of real numbers. An operator ψ on $\mathcal{L}$ is called **increasing** if it preserves the partial ordering, i.e., $f \leq g$ implies $\psi(f) \leq \psi(g)$. Increasing operators are of great importance, and among them four fundamental examples are:

$$\delta \text{ is } \textbf{dilation} \iff \delta(\bigvee_{i \in I} f_i) = \bigvee_{i \in I} \delta(f_i) \tag{17}$$

$$\varepsilon \text{ is } \textbf{erosion} \iff \varepsilon(\bigwedge_{i \in I} f_i) = \bigwedge_{i \in I} \varepsilon(f_i) \tag{18}$$

$$\alpha \text{ is } \textbf{opening} \iff \alpha \text{ is increasing, idempotent, and anti-extensive} \tag{19}$$

$$\beta \text{ is } \textbf{closing} \iff \beta \text{ is increasing, idempotent, and extensive} \tag{20}$$

where I is an arbitrary index set, idempotence means that $\alpha(\alpha(f)) = \alpha(f)$, and (anti-)extensivity of $(\alpha)\beta$ means that $\alpha(f) \leq f \leq \beta(f)$ for all f.

The above definitions allow broad classes of signal operators to be grouped as lattice dilations, or erosions, or openings, or closings and their common properties to be studied under the unifying lattice framework. Thus, the translation-invariant morphological dilations $\oplus$, erosions $\ominus$, openings $\circ$, and closings $\bullet$ are simple special cases of their lattice counterparts.

3 Morphological Filters for Enhancement

3.1 Image Smoothing or Simplification

3.1.1 Lattice Opening Filters

The three types of nonlinear filters defined below are lattice openings in the sense of (19) and have proven to be very useful for image enhancement.

If a 2D image f contains 1D objects, e.g. lines, and B is a 2D disk-like structuring element, then the simple opening or closing of f by B will eliminate these 1D objects. Another problem arises when f contains large-scale objects with sharp corners that need to be preserved; in such cases opening or closing f by a disk B will round these corners. These two problems could be avoided in some cases if we replace the conventional opening with a **radial opening**

$$\alpha(f) = \bigvee_{\theta} f \circ L_\theta \tag{21}$$

where the sets L_θ are rotated versions of a line segment L at various angles $\theta \in [0, 2\pi)$. This has the effect of preserving an object in f if this object is left unchanged after the opening by L_θ in at least one of the possible orientations θ. See Fig. 1 for examples.

There are numerous image enhancement problems where what is needed is suppresion of arbitrarily-shaped connected components in the input image whose areas (number of pixels) are smaller than a certain threshold n. This can be accomplished by the **area opening** of size n which, for binary images, keeps only the connected components whose area is $\geq n$ and eliminates the rest. The area opening can also be extended to graylevel images.

Consider now a set $X = \bigcup_i X_i$ as a union of disjoint connected components X_i and let $M \subseteq X_j$ be a *marker* in the j-th component; i.e., M could be a single point or some feature set in X that lies only in X_j. Let us define the **opening by reconstruction** as the operator

$$\mathrm{MR}_X(M) \equiv \text{connected component of } X \text{ containing } M. \tag{22}$$

This is a lattice opening that from the input set M yields as output exactly the component X_j containing the marker. Its output is called the *morphological reconstruction* of the component from the marker. It can extract large-scale components of the image from knowledge only of a smaller marker inside them. An algorithm to implement the opening by reconstruction is based on the *conditional dilation* of M by B within X:

$$\delta_{B|X}(M) \equiv (M \oplus B) \cap X \tag{23}$$

If B is a disk with a radius smaller than the distance between X_j and any of the other components, then by iterating this conditional dilation we can obtain in the limit

$$\lim_{n\to\infty} \underbrace{(\delta_{B|X} ... (\delta_{B|X}(\delta_{B|X}(M))))}_{n \text{ times}} = \mathrm{MR}_X(M)$$

the whole component X_j. Replacing the binary with graylevel images, the set dilation with function dilation, and $\cap$ with $\wedge$ yields the graylevel opening by reconstruction. Openings (and closings) by reconstruction have proven to be extremely useful for image simplification because they can suppress small features and keep only large-scale objects without any smoothing of their boundaries. Examples are shown in Fig. 1.

3.1.2 Multiscale Morphological Smoothers

Multiscale image analysis has recently emerged as a useful framework for many computer vision and image processing tasks, including (i) noise suppression at various scales and (ii) feature detection at large scales followed by refinement of their location or value at smaller scales. Most of the previous work in this area was based on a *linear* multiscale smoothing, i.e., convolutions with a Gaussian with a variance proportional to scale. However, these linear smoothers blur or shift image edges, as shown in Fig. 1. In contrast, there is a variety of *nonlinear* smoothing filters, including the morphological openings and closings that can provide a multiscale image ensemble [17, 8] and avoid the above shortcomings of linear smoothers. For example, Fig. 1 shows three types of *clos-openings* (i.e., cascades of openings followed by closings): 1) Flat clos-openings by a 2D disk-like

structuring element which preserve the vertical image edges but may distort horizontal edges by fitting the shape of the structuring element. 2) Radial flat clos-openings that preserve both the vertical edges as well as any line features along the directions (0^o, 45^o, 90^o, 135^o) of the four line segments used as structuring elements. 3) Gray-level clos-openings by reconstruction which are especially useful because they can extract the exact outline of a certain object by locking on it while smoothing out all its surroundings; the marker for the opening (closing) by reconstruction was an erosion (dilation) of the original image by a disk of radius equal to scale.

The required building blocks for the above morphological smoothers are the multiscale dilations and erosions. The simplest multiscale dilation and erosion of an image $f(x)$ at scales $t > 0$ are the flat dilations/erosions of f by scaled versions $tB = \{tz : z \in B\}$ of a unit-scale planar compact convex set B (e.g., a disk, a rhombus, a square)

$$\delta(x,t) \equiv (f \oplus tB)(x) \;\; , \;\; \varepsilon(x,t) \equiv (f \ominus tB)(x) \tag{24}$$

which apply both to graylevel and binary images. One discrete approach to implement multiscale dilations and erosions is to use scale recursion, i.e., $f \oplus (n+1)B = (f \oplus nB) \oplus B$ where $n = 0, 1, 2, ...$, and nB denotes the n-fold dilation of B with itself. An alternative and more recent approach that uses continuous models for multiscale smoothing is based on *partial differential equations (PDEs)*. This was inspired by the modeling of linear multiscale image smoothing via the isotropic heat diffusion PDE $\partial U/\partial t = \nabla^2 U$, where $U(x,t)$ is the convolution of the initial image $f(x) = U(x,0)$ with a Gaussian at scale t. Similarly, the multiscale dilation $\delta(x,t)$ of f by a disk of radious (scale) t can be gererated as a weak solution of the following nonlinear PDE

$$\frac{\partial \delta}{\partial t} = ||\nabla \delta|| \tag{25}$$

with initial condition $\delta(x,0) = f(x)$, where ∇ denotes the spatial gradient operator and $||\cdot||$ is the Euclidean norm. The generating PDE for the erosion is $\partial \varepsilon/\partial t = -||\nabla \varepsilon||$. A review and references of the PDE approach to multiscale morphology can be found in [6]. In general, the PDE approach yields very close approximations to Euclidean multiscale morphology with arbitrary subpixel accuracy.

3.1.3 Noise Suppresion by Median and Alternating Sequential Filters

In their behavior as nonlinear smoothers, as shown in Fig. 2, the medians act similarly to an *open-closing* $(f \circ B) \bullet B$ by a convex set B of diameter about half the diameter of the median window [7]. The open-closing has the advantages over the median that it requires less computation and decomposes the noise suppression task into two independent steps, i.e., suppressing positive spikes via the opening and negative spikes via the closing.

The popularity and efficiency of the simple morphological openings and closings to suppress impulse noise is supported by the following theoretical development [19]. Assume a class of sufficiently smooth random input images which is the collection of all subsets of a finite mask W that are open (or closed) with respect to a set B and assign a uniform probability distribution on this collection. Then, a discrete binary input image X is a random realization from this collection; i.e., use ideas from random sets [17] to model X. Further, X is corrupted by a union (or intersection) noise N

which is a 2D sequence of i.i.d. binary Bernoulli random variables with probability $p \in [0, 1)$ of occurrence at each pixel. The observed image is the noisy version $Y = X \cup N$ (or $Y = X \cap N$). Then, the maximum-a-posteriori estimate [19] of the original X given the noisy image Y is the opening (or closing) of the observed Y by B.

Another useful generalization of openings and closings involves cascading open-closings $\beta_t\alpha_t$ at multiple scales $t = 1, ..., r$, where $\alpha_t(f) = f \circ tB$ and $\beta_t(f) = f \bullet tB$. This generates a class of efficient nonlinear smoothing filters

$$\psi_{asf}(f) = \beta_r\alpha_r ... \beta_2\alpha_2\beta_1\alpha_1(f) \tag{26}$$

called **alternating sequential filters**, which smooth progressively from the smallest scale possible up to a maximum scale r and have a broad range of applications [18]. Their optimal design is addressed in [16].

3.2 Edge or Contrast Enhancement

3.2.1 Morphological Gradients

Consider the difference between the flat dilation and erosion of an image f by a symmetric disk-like set B containing the origin whose diameter $\mathrm{diam}(B)$ is very small:

$$\mathrm{edge}(f) = \frac{(f \oplus B) - (f \ominus B)}{\mathrm{diam}(B)} \tag{27}$$

If f is binary, $\mathrm{edge}(f)$ extracts its boundary. If f is gray-level, the above residual enhances its edges [9, 17] by yielding an approximation to $||\nabla f||$, which is obtained in the limit of (27) as $\mathrm{diam}(B) \to 0$. See Fig. 3. Further, thresholding this 'morphological gradient' leads to binary edge detection.

The symmetric morphological gradient (27) is the average of two asymmetric ones: the erosion gradient $f - (f \ominus B)$ and the dilation gradient $(f \oplus B) - f$. The symmetric or asymmetric morphological edge-enhancing gradients can be made more robust for edge detection by first smoothing the input image with a linear blur [4]. These hybrid edge detection schemes that largely contain morphological gradients are computationally more efficient and perform comparably or in some cases better than several conventional schemes based only on linear filters.

3.2.2 Toggle Contrast Filter

Consider a graylevel image $f[x]$ and a small-size symmetric disk-like structuring element B containing the origin. The following discrete nonlinear filter [3] can enhance the local contrast of f by sharpening its edges:

$$\psi(f)[x] = \begin{cases} (f \oplus B)[x] & \text{if} \quad (f \oplus B)[x] - f[x] \le f[x] - (f \ominus B)[x] \\ (f \ominus B)[x] & \text{if} \quad (f \oplus B)[x] - f[x] > f[x] - (f \ominus B)[x] \end{cases} \tag{28}$$

At each pixel x, the output value of this filter *toggles* between the value of the dilation of f by B (i.e., the maximum of f inside the moving window B centered) at x and the value of its erosion by B (i.e., the minimum of f within the same window) according to which is closer to the input

value $f[x]$. The toggle filter is usually applied not only once but is *iterated*. The more iterations, the more contrast enhancement. Further, the iterations converge to a *limit (fixed point)* [3] reached after a finite number of iterations. Examples are shown in Fig. 4 and Fig. 5.

As discussed in [15, 6], the above discrete toggle filter is closely related to the operation and numerical algorithm behind a nonlinear (shock-wave) PDE proposed in [10] to deblur images and/or enhance their contrast by edge sharpening. For 1D images such a PDE is

$$\frac{\partial u}{\partial t} = -\left|\frac{\partial u}{\partial x}\right| \operatorname{sign}\left(\frac{\partial^2 u}{\partial x^2}\right) \tag{29}$$

Starting at $t = 0$, with the blurred image $u(x, 0) = f(x)$ as the initial data, and running the numerical algorithm implementing this PDE until some time t yields a filtered image $u(x, t)$. Its goal is to restore blurred edges sharply, accurately and in a nonoscillatory way by propagating shocks (i.e., discontinuities in the signal derivatives). Steady state is reached as $t \to \infty$. Over convex regions ($\partial^2 u/\partial x^2 > 0$) this PDE acts as a 1D erosion PDE $\partial u/\partial t = -|\partial u/\partial x|$ which models multiscale erosion of $f(x)$ by the horizontal line segment $[-t, t]$ and shifts parts of the graph of $u(x, t)$ with positive (negative) slope to the right (left) but does not move the extrema or inflection points. Over concave regions ($\partial^2 u/\partial x^2 < 0$) it acts as a 1D dilation PDE $\partial u/\partial t = |\partial u/\partial x|$ which models multiscale dilation of $f(x)$ by the same segment and reverses the direction of propagation. For certain piecewise-constant signals blurred via linear convolution with finite-window smooth tapered symmetric kernels, the shock filtering $u(x, 0) \mapsto u(x, \infty)$ can recover the original signal and thus achieve an exact deconvolution [10]; an example of such a case is shown in Fig. 4.

4 Morphological Filters for Detection

4.1 Morphological Correlation

Consider two real-valued discrete image signals $f[x]$ and $g[x]$. Assume that g is a signal pattern to be found in f. To find which shifted version of g "best" matches f a standard approach has been to search for the shift lag y that minimizes the *mean squared error* $E_2[y] = \sum_{x \in W} (f[x+y] - g[x])^2$ over some subset W of $\mathbb{Z}^2$. Under certain assumptions, this matching criterion is equivalent to maximizing the *linear cross-correlation* $L_{fg}[y] \equiv \sum_{x \in W} f[x+y]g[x]$ between f and g. A discussion of linear template matching can be found in Chapter 3.1.

Although less mathematical tractable than the mean squared error criterion, a statistically more robust criterion is to minimize the *mean absolute error*

$$E_1[y] = \sum_{x \in W} |f[x+y] - g[x]|$$

This mean absolute error criterion corresponds to a nonlinear signal correlation used for signal matching; see [8] for a review. Specifically, since $|a - b| = a + b - 2\min(a, b)$, under certain assumptions (e.g., if the error norm and the correlation is normalized by dividing it with the average area under the signals f and g), minimizing $E_1[y]$ is equivalent to maximizing the *morphological* cross-correlation

$$M_{fg}[y] \equiv \sum_{x \in W} \min(f[x+y], g[x]) \tag{30}$$

It can be shown experimentally and theoretically that the detection of g in f is indicated by a sharper matching peak in $M_{fg}[y]$ than in $L_{fg}[y]$. In addition, the morphological (sum of minima) correlation is faster than the linear (sum of products) correlation. These two advantages of the morphological correlation coupled with the relative robustness of the mean absolute error criterion make it promising for general signal matching.

4.2 Binary Object Detection and Rank Filtering

Let us approach the problem of binary image object detection in the presence of noise from the viewpoint of statistical hypothesis testing and rank filtering. Assume that the observed discrete binary image $f[x]$ within a mask W has been generated under one of the following two probabilistic hypotheses:

$$\begin{aligned} H_0: &\quad f[x] = e[x], \;\; x \in W. \\ H_1: &\quad f[x] = |g[x-y] - e[x]|, \;\; x \in W. \end{aligned}$$

Hypothesis H_1 (H_0) stands for 'object present' ('object not present') at pixel location y. The *object* $g[x]$ is a deterministic binary template. The *noise* $e[x]$ is a stationary binary random field which is a 2D sequence of independent identically distributed (i.i.d.) random variables taking value 1 with probability p and 0 with probability $1-p$, where $0 < p < 0.5$. The *mask* $W = G_{+y}$ is a finite set of pixels equal to the region G of support of g shifted to location y at which the decision is taken. (For notational simplicity, G is assumed to be symmetric, i.e., $G = G^r$.) The absolute-difference superposition between g and e under H_1 forces f to always have values 0 or 1. Intuitively, such a signal/noise superposition means that the noise e toggles the value of g from 1 to 0 and from 0 to 1 with probability p at each pixel. This noise model can be viewed either as the common binary symmetric channel noise in signal transmission or as a binary version of the salt-and-pepper noise. To decide whether the object g occurs at y we use a Bayes decision rule that minimizes the total probability of error and hence leads to the *likelihood ratio test*

$$\frac{Pr(f/H_1)}{Pr(f/H_0)} \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} \frac{Pr(H_0)}{Pr(H_1)} \tag{31}$$

where $Pr(f/H_i)$ are the likelihoods of H_i with respect to the observed image f, and $Pr(H_i)$ are the *a priori* probabilities. This is equivalent to

$$M_{fg}[y] = \sum_{x \in W} \min(f[x], g[x-y]) \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} \theta = \frac{1}{2}\left(\frac{\log[Pr(H_0)/Pr(H_1)]}{\log[(1-p)/p]} + \mathrm{card}(G) \right) \tag{32}$$

Thus, the selected statistical criterion and noise model lead to compute the morphological (or equivalently linear) binary correlation between a noisy image and a known image object and compare it to a threshold for deciding whether the object is present.

Thus, optimum detection in a binary image f of the presence of a binary object g requires comparing the binary correlation between f and g to a threshold θ. This is equivalent[5] to performing a r-th rank filtering on f by a set G equal to the support of g, where $1 \leq r \leq \text{card}(G)$ and r is related to θ. Thus, the rank r reflects the area portion of (or a probabilistic confidence score for) the shifted template existing around pixel y. For example, if $Pr(H_0) = Pr(H_1)$, then $r = \theta = \text{card}(G)/2$ and hence the binary median filter by G becomes the optimum detector.

4.3 Hit-Miss Filter

The set erosion (3) can also be viewed as Boolean template matching since it gives the center points at which the shifted structuring element fits inside the image object. If we now consider a set A probing the image object X and another set B probing the background X^c, the set of points at which the shifted pair (A, B) fits inside the image X is the **hit-miss transformation** of X by (A, B):

$$X \otimes (A, B) \equiv \{x : A_{+x} \subseteq X,\ B_{+x} \subseteq X^c\} \tag{33}$$

In the discrete case, this can be represented by a Boolean product function whose uncomplemented (complemented) variables correspond to points of A (B). It has been used extensively for binary feature detection [17]. It can actually model all binary template matching schemes in binary pattern recognition that use a pair of a positive and a negative template [13].

In the presence of noise, the hit-miss filter can be made more robust by replacing the erosions in its definitions with rank filters that do not require an exact fitting of the whole template pair (A, B) inside the image but only a part of it.

4.4 Morphological Peak/Valley Feature Detection

Residuals between openings or closings and the original image offer an intuitively simple and mathematically formal way for peak or valley detection. Specifically, subtracting from an input image f its opening by a compact convex set B yields an output consisting of the image peaks whose support cannot contain B. This is the *top-hat transformation* [9]

$$\text{peak}(f) = f - (f \circ B) \tag{34}$$

that has found numerous applications in geometric feature detection [17]. It can detect *bright blobs*, i.e., regions with significantly brighter intensities relative to the surroundings. The shape of the detected peak's support is controlled by the shape of B, whereas the scale of the peak is controlled by the size of B. Similarly, to detect *dark blobs*, modeled as image intensity valleys, we can use the valley detector

$$\text{valley}(f) = (f \bullet B) - f \tag{35}$$

See Fig. 3 for examples.

[5] An alternative implementation and view of binary rank filtering is via *thresholded convolutions*, where a binary image is linearly convolved with the indicator function of a set G with $n = \text{card}(G)$ pixels and then the result is thresholded at an integer level r between 1 and n; this yields the output of the r-th rank filter by G acting on the input image.

The morphological peak/valley detectors are simple, efficient, and have some advantages over curvature-based approaches. Their applicability in situations where the peaks or valleys are not clearly separated from their surroundings is further strengthened by generalizing them in the following way. The conventional opening in (34) is replaced by a general lattice opening such as an area opening or opening by reconstruction. This generalization allows a more effective estimation of the image background surroundings around the peak and hence a better detection of the peak.

5 Optimal Design of Morphological Filters for Enhancement

5.1 Brief Survey of Existing Design Approaches

Morphological and rank/stack filters are useful for image enhancement and are closely related since they can all be represented as maxima of morphological erosions [7]. Despite the wide application of these nonlinear filters, very few ideas exist for their optimal design. The current four main approaches are: (a) designing morphological filters as a finite union of erosions [5] based on the morphological basis representation theory (outlined in Section 2.3); (b) designing stack filters via threshold decomposition and linear programming [1]; (c) designing morphological networks using either voting logic and rank tracing learning or simulated annealing [20]; (d) designing morphological/rank filters via a gradient-based adaptive optimization [14]. Approach (a) is limited to binary increasing filters. Approach (b) is limited to increasing filters processing nonnegative quantized signals. Approach (c) needs a long time to train and convergence is complex. In contrast, approach (d) is more general since it applies to both increasing and non-increasing filters and to both binary and real-valued signals. The major difficulty involved is that rank functions are *not* differentiable, which imposes a deadlock on how to adapt the coefficients of morphological/rank filters using a gradient-based algorithm. The methodology described in this section is an extension and improvement to the design methodology (d), leading to a new approach that is simpler, more intuitive and numerically more robust.

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 (rank-type) sub-systems, have frequently been proposed in the research literature. A typical example is the class of L-filters that are linear combinations of rank filters. Several adaptive algorithms have also been developed for their design, which illustrated the potential of adaptive hybrid filters for image processing applications, especially in the presence of non-Gaussian noise.

Given the applicability of hybrid systems and the relatively few existing ideas to design their nonlinear part, in this section we present a general class of nonlinear systems, called **morphological/rank/linear (MRL)** filters [11], that contains as special cases morphological, rank, and linear filters, and we develop an efficient method for their adaptive optimal design. MRL filters consist of a linear combination between a morphological/rank filter and a linear FIR filter. 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.

5.2 MRL Filters

We shall use a vector notation to represent the values of the 1D or 2D sampled signal (after some enumeration of the signal samples) inside an n-point moving window. Let $\underline{x} = (x_1, x_2, \cdots, x_n)$ in $\mathbb{R}^n$ represent the input signal segment and y be the output value from the filter. The MRL filter is defined as the shift-invariant system whose local signal transformation rule $\underline{x} \mapsto y$ is given by

$$\begin{aligned} y &\equiv \lambda\alpha + (1-\lambda)\beta \ , \\ \alpha &= \mathcal{R}_r(\underline{x} + \underline{a}) = \mathcal{R}_r(x_1 + a_1, x_2 + a_2, \cdots, x_n + a_n) \ , \\ \beta &= \underline{x} \cdot \underline{b}^T = x_1 b_1 + x_2 b_2 + \cdots + x_n b_n \ , \end{aligned} \tag{36}$$

where $\lambda \in \mathbb{R}$, $\underline{a}, \underline{b} \in \mathbb{R}^n$, and '$(\cdot)^T$' denotes transposition. $\mathcal{R}_r(\underline{t})$ is the r-th rank function of $\underline{t} \in \mathbb{R}^n$. It is evaluated by sorting the components of $\underline{t} = (t_1, t_2, \cdots, t_n)$ in decreasing order, $t_{(1)} \geq t_{(2)} \geq \cdots \geq t_{(n)}$, and picking the r-th element of the sorted list; *i.e.*, $\mathcal{R}_r(\underline{t}) \equiv t_{(r)}$, $r = 1, 2, \cdots, n$. 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. 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 decreases. From (36) it follows that, computing each output sample requires $2n + 1$ additions, $n + 2$ multiplications and an n-point sorting operation.

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'. We define the unit sample function $q(v)$, $v \in \mathbb{R}$, as

$$q(v) \equiv \begin{cases} 1 & , \text{ if } v = 0 \\ 0 & , \text{ otherwise} \end{cases} \tag{37}$$

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

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

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 r-th **rank indicator vector** $\underline{c}$ of $\underline{t}$ is defined by

$$\underline{c}(\underline{t}, r) \equiv \frac{Q(z\underline{1} - \underline{t})}{Q(z\underline{1} - \underline{t}) \cdot \underline{1}^T} \ , \quad z = \mathcal{R}_r(\underline{t}) \ , \tag{38}$$

where $\underline{1} = (1, 1, \cdots, 1)$. Thus, the rank indicator vector marks the locations in $\underline{t}$ where the z value occurs. It has many interesting properties [11], which include the following. It has unit area:

$$\underline{c} \cdot \underline{1}^T = 1$$

It yields an inner-product representation of the rank function:

$$\underline{c} \cdot \underline{t}^T = \mathcal{R}_r(\underline{t})$$

Further, for r fixed, if $\underline{c}$ is constant in a neighborhood of some $\underline{t}_0$, then the r-th rank function $\mathcal{R}_r(\underline{t})$ is differentiable at $\underline{t}_0$ and

$$\left.\frac{\partial \mathcal{R}_r(\underline{t})}{\partial \underline{t}}\right|_{\underline{t}=\underline{t}_0} = \underline{c}(\underline{t}_0, r). \tag{39}$$

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

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 a **smoothed impulse** $q_\sigma(v)$ that depends on a scale parameter $\sigma \geq 0$ and has at least the following required properties:

$$\begin{aligned} &q_\sigma(v) = q_\sigma(-v) && \text{(symmetry)} \\ &q_\sigma(v) \to q(v)\ \forall\, v && \text{as} \quad \sigma \to 0, \\ &q_\sigma(v) \to 1\ \forall\, v && \text{as} \quad \sigma \to \infty. \end{aligned} \tag{40}$$

Functions like $\exp\left[-\frac{1}{2}(v/\sigma)^2\right]$ or $\mathrm{sech}^2(v/\sigma)$ are natural choices for $q_\sigma(v)$.

From the filter definition (36), 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

$$r \equiv \left\lfloor n - \frac{n-1}{1+\exp(-\rho)} + 0.5 \right\rfloor,\ \rho \in \mathbb{R}\ , \tag{41}$$

where $\lfloor \cdot + 0.5 \rfloor$ denotes the usual rounding operation and n is the dimension of the input signal vector $\underline{x}$ inside the moving window. Thus, the weight vector to be used in the filter design task is defined by

$$\underline{w} \equiv (\underline{a},\ \rho,\ \underline{b},\ \lambda)\ , \tag{42}$$

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

5.3 LMS Approach to Designing Optimal MRL Filters

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

$$\underline{w}(i+1) = \underline{w}(i) - \mu_0 \nabla J(\underline{w})|_{\underline{w}=\underline{w}(i)}\ , \tag{43}$$

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 (43) 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 i-th update $\underline{w}(i)$ of the weight vector, we use

$$J(\underline{w}(i)) = \frac{1}{M} \sum_{k=i-M+1}^{i} e^2(k) \ , \tag{44}$$

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

$$e(k) = d(k) - y(k) \tag{45}$$

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 least mean square (LMS)* algorithm, is

$$\underline{w}(i+1) = \underline{w}(i) + \frac{\mu}{M} \sum_{k=i-M+1}^{i} e(k) \frac{\partial y(k)}{\partial \underline{w}} \bigg|_{\underline{w}=\underline{w}(i)} \ , \ i = 0, 1, 2, \cdots , \tag{46}$$

where $\mu = 2\mu_0$. From (42) and (36)

$$\frac{\partial y}{\partial \underline{w}} = \left(\frac{\partial y}{\partial \underline{a}}, \ \frac{\partial y}{\partial \rho}, \ \frac{\partial y}{\partial \underline{b}}, \ \frac{\partial y}{\partial \lambda} \right) \ = \ \left[\lambda \frac{\partial \alpha}{\partial \underline{a}}, \ \lambda \frac{\partial \alpha}{\partial \rho}, \ (1-\lambda)\underline{x}, \ \alpha - \beta \right] . \tag{47}$$

According to (39) and our design choice, we set

$$\frac{\partial \alpha}{\partial \underline{a}} = \underline{c} = \frac{Q(\alpha \underline{1} - \underline{x} - \underline{a})}{Q(\alpha \underline{1} - \underline{x} - \underline{a}) \cdot \underline{1}^T} \ , \quad \alpha = \mathcal{R}_r(\underline{x} + \underline{a}). \tag{48}$$

The final unknown is $s = \partial\alpha/\partial\rho$, which will be one more design choice. Notice from (41) and (36) 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

$$\frac{\partial \alpha}{\partial \rho} = s \equiv 1 - \frac{1}{n} Q(\alpha \underline{1} - \underline{x} - \underline{a}) \cdot \underline{1}^T \ , \quad \alpha = \mathcal{R}_r(\underline{x} + \underline{a}) \ , \tag{49}$$

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

Finally, to improve the numerical robustness of the training algorithm, we will frequently replace the unit sample function by smoothed impulses (obeying (40)), 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 (46). We should use small values of σ such that $q_\sigma(v)$ is close enough to $q(v)$. A possible systematic way to select the smoothing parameter σ could be to set $|q_\sigma(v)| \leq \epsilon$ for $|v| \geq \delta$, so that, for some desired ϵ and δ, $\sigma = \delta / \sqrt{\ln(1/\epsilon^2)}$.

Theoretical conditions for convergence of the training process (46) can be derived under the following considerations. The goal is to find upper bounds μ_w to the step size μ, such that (46) can

converge if $0 < \mu < \mu_w$. 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 (46) to track only the fourth unknown parameter (set) of $\underline{w}^*$ in a noiseless environment. If the training process (46) is convergent, then $\lim_{i\to\infty} \|\underline{w}(i) - \underline{w}^*\| = 0$, where $\|\cdot\|$ is some error norm. By analyzing the behavior of $\|\underline{w}(i) - \underline{w}^*\|$, under the above assumptions, conditions for convergence have been found in [11].

5.4 Application of Optimal MRL Filters to Enhancement

The proper operation of the training process (46) has been verified in [11] through experiments confirming that, if the conditions for convergence are met, our design algorithm converges fast to the real parameters of the MRL-filter within small error distances.

We illustrate its applicability to image enhancement via an experiment.[6] 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% multi-valued 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 (46) 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 non-flat median filter and a linear FIR filter, where some elements of $\underline{a}$ and $\underline{b}$ present more influence in the filtering process.

Figure 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

[6] Implementation details: The images are scanned twice during the training process, following a zig zag path from top to bottom, and then from bottom to top. The local input vector $\underline{x}$ is obtained at each pixel via column-by-column indexing of the image values inside a n-point square window centered around the pixel. The vectors $\underline{a}$ and $\underline{b}$ are indexed the same way. The unit sample function $q(v)$ is approximated by $q_\sigma(v) = \exp\left[-\frac{1}{2}(v/\sigma)^2\right]$, with $\sigma = 0.001$. The image values are normalized to be in the range $[0, 1]$.

about 3 dB. Spatial error plots are also included which show that, the optimal MRL-filter preserves better the image structure since its corresponding spatial error is more uncorrelated than the error of the median filter.

For the type of noise used in this experiment, we must have at least part of the original (noiseless) image, otherwise we would not be able to provide a good estimate to the optimal filter parameters during the training process (46). In order to validate this point, we repeated the above experiment using 100x100 sub-images of the training image (only 17% of the pixels), and the resulting MRL-filter still outperformed the median filter by about 2.3dB. 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 [14]. But this approach is only appropriate for certain types of impulse noise.

An exhaustive comparison of different filter structures for noise cancellation is beyond the scope of this chapter. Nevertheless, this experiment was extended with the adaptive design of a 3x3 L-filter 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 0.2dB better than the (flat) median filter.

Acknowledgement

Part of this chapter dealt with the authors' research work which was supported by the US National Science Foundation under Grants MIPS-86-58150 and MIP-94-21677.

References

[1] 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.

[2] H.J.A.M. Heijmans, *Morphological Image Operators*, Acad. Press, Boston, 1994.

[3] H. P. Kramer and J. B. Bruckner, "Iterations of a Nonlinear Transformation for Enhancement of Digital Images", *Pattern Recognition*, vol. 7, pp. 53–58, 1975.

[4] J.S.J. Lee, R.M. Haralick and L.G. Shapiro, "Morphologic Edge Detection," *IEEE Trans. Rob. Autom.*, vol. RA-3, pp. 142-156, Apr. 1987.

[5] R. P. Loce and E. R. Dougherty, "Facilitation of Optimal Binary Morphological Filter Design via Structuring Element Libraries and Design Constraints", *Optical Engineering*, vol. 31, pp. 1008-1025, May 1992.

[6] P. Maragos, "Partial Differential Equations in Image Analysis: Continuous Modeling, Discrete Processing", *Proc. 1998 European Signal Processing Conference (EUSIPCO)*, Rhodes, Greece. Published in: *Signal Processing IX: Theories and Applications*, vol. II, pp. 527–536, EURASIP Press, 1998.

[7] P. Maragos and R. W. Schafer, "Morphological Filters. Part I: Their Set-Theoretic Analysis and Relations to Linear Shift-Invariant Filters. Part II: Their Relations to Median, Order-Statistic, and Stack Filters," *IEEE Trans. Acoust. Speech, Signal Process.*, vol. 35, pp.1153–1184, Aug. 1987; *ibid*, vol. 37, p. 597, Apr. 1989.

[8] P. Maragos and R. W. Schafer, "Morphological Systems for Multidimensional Signal Processing", *Proc. IEEE*, vol. 78, pp. 690–710, April 1990.

[9] F. Meyer, "Contrast Feature Extraction", *Proc. 1977 European Symp. on Quantitative Analysis of Microstructures in Materials Science, Biology and Medicine*, France. Published in: *Special Issues of Practical Metallography*, J.L. Chermant, ed., Riederer-Verlag, Stuttgart, 1978, pp. 374-380.

[10] S. Osher and L. I. Rudin, "Feature-Oriented Image Enhancement Using Schock Filters", *SIAM J. Numer. Anal.*, vol. 27, pp. 919–940, Aug. 1990.

[11] L. F. C. Pessoa and P. Maragos, "MRL-Filters: A General Class of Nonlinear Systems and Their Optimal Design for Image Processing," *IEEE Trans. Image Processing*, vol. 7, pp. 966–978, July 1998.

[12] K. Preston, Jr., and M.J.B. Duff, *Modern Cellular Automata*, Plenum Press, 1984.

[13] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Vols. 1 & 2, Acad. Press, NY, 1982.

[14] P. Salembier, "Adaptive Rank Order Based Filters," *Signal Processing*, vol. 27, pp. 1–25, 1992.

[15] J.G.M. Schavemaker, M.J.T. Reinders and R. Van den Boomgaard, "Image Sharpening by Morphological Filtering", *Proc. IEEE Workshop on Nonlinear Signal & Image Processing*, MacKinac Island, Michigan, Sep. 1997.

[16] D. Schonfeld and J. Goutsias, "Optimal Morphological Pattern Restoration from Noisy Binary Images", *IEEE Trans. Pattern Anal. Machine Intellig.*, vol. 13, pp. 14-29, Jan. 1991.

[17] J. Serra, *Image Analysis and Mathematical Morphology*, Acad. Press, NY, 1982.

[18] J. Serra, ed., *Image Analysis and Mathematical Morphology, Vol.2: Theoretical Advances*, Acad. Press, NY, 1988.

[19] N. D. Sidiropoulos, J. S. Baras and C. A. Berenstein, "Optimal Filtering of Digital Binary Images Corrupted by Union/Intersection Noise", *IEEE Trans. Image Processing*, vol. 3, pp. 382–403, July 1994.

[20] S. S. Wilson, "Training Structuring Elements in Morphological Networks", in *Mathematical Morphology in Image Processing*, E.R. Dougherty, ed., Marcel Dekker, NY, 1993.

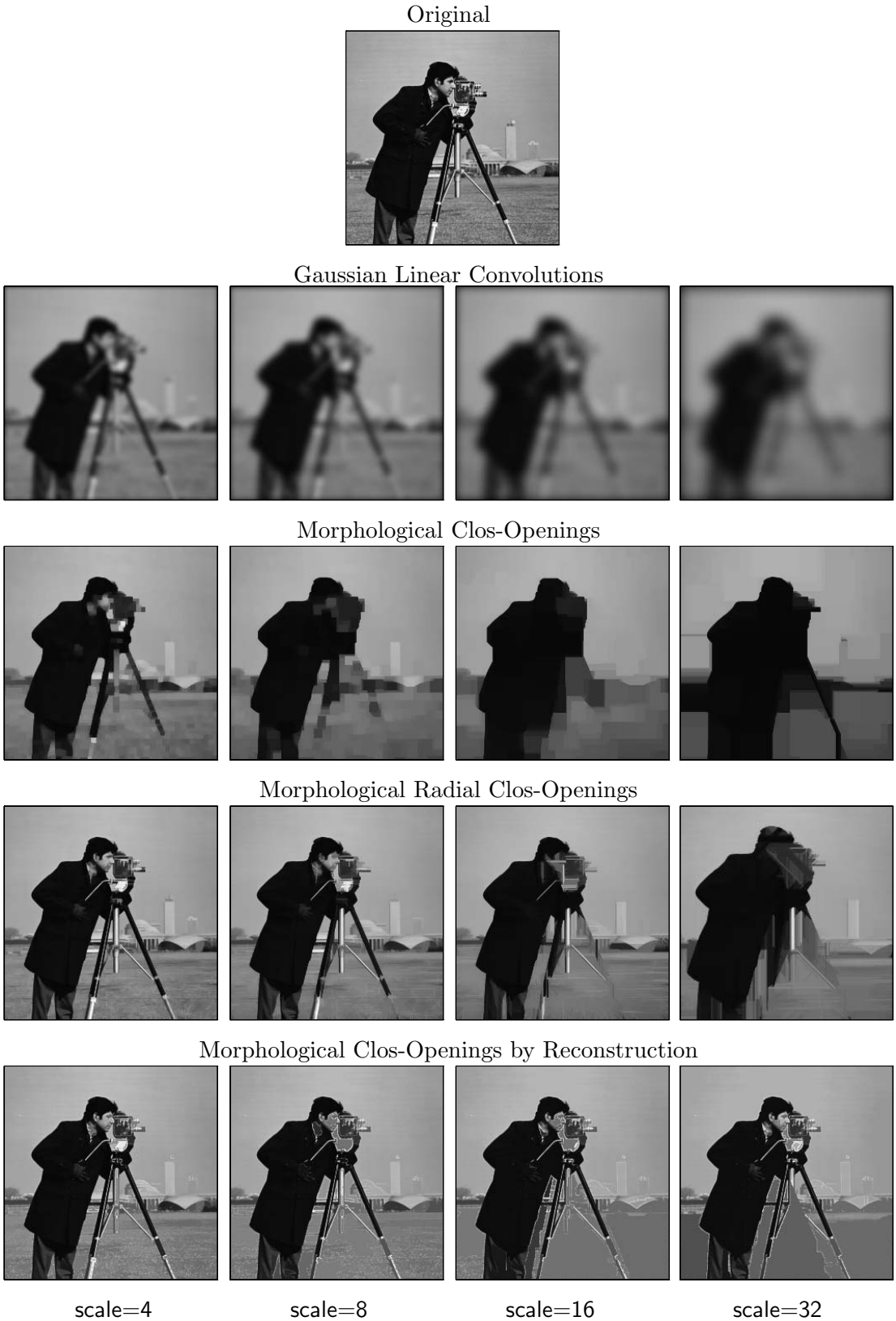
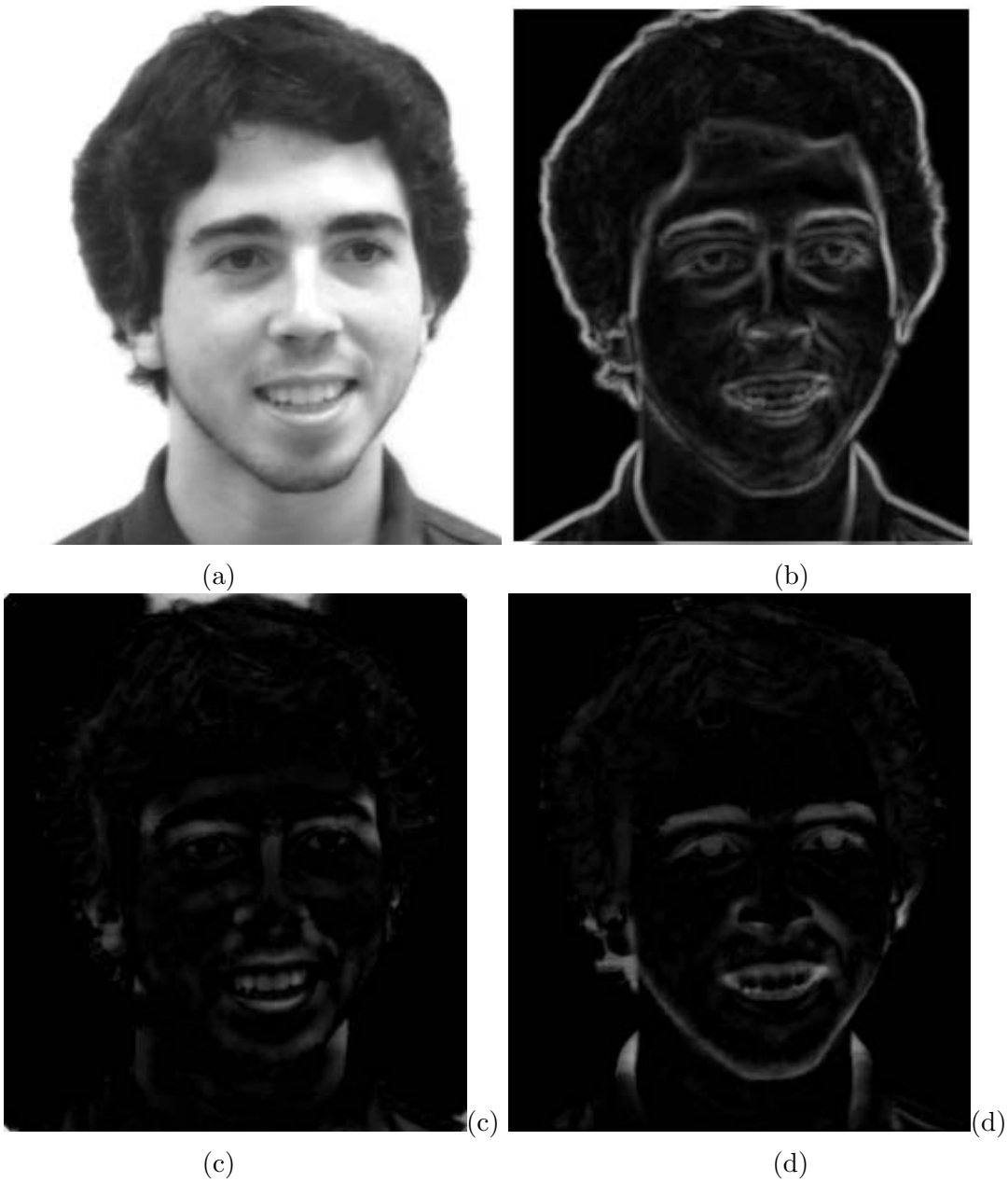


Figure 1: Linear and morphological multiscale image smoothers. (The scale parameter was defined as: the variance of the Gaussians for linear convolutions; the radius of the structuring element for clos-openings; the scale of the marker for the reconstruction filters.)

(a) (b)

(c) (d)

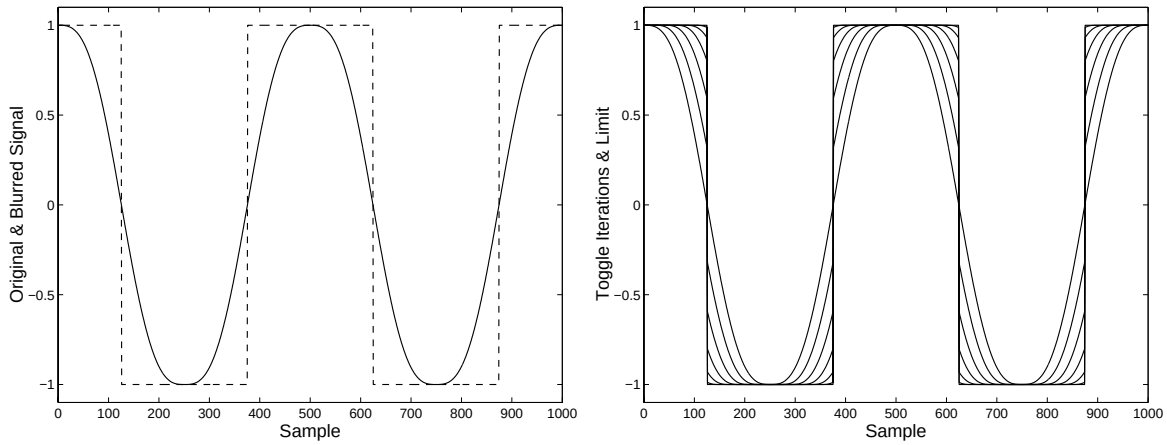
Figure 2: (a) Original clean image. (b) Noisy image obtained by corrupting the original with two-level salt-and-pepper noise occuring with probability 0.1 (PSNR=18.9 dB). (c) Open-closing of noisy image by a 2×2-pel square (PSNR=25.4 dB). (d) Median of noisy image by a 3×3-pel square (PSNR=25.4 dB).



(a) (b)

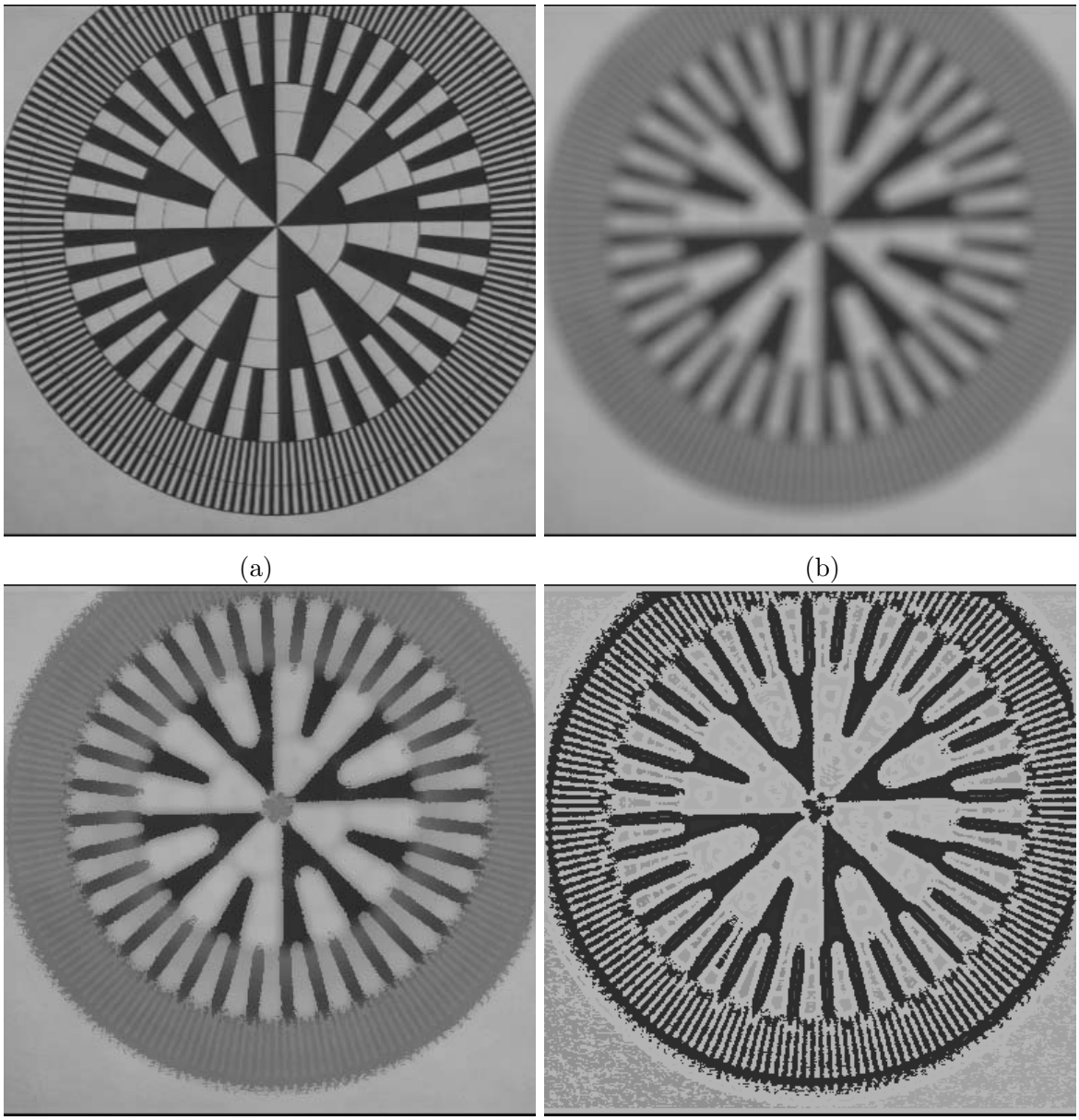
(c) (d)

Figure 3: Morphological edge and blob detectors. (a) Image f. (b) Edges: morphological gradient $f \oplus B - f \ominus B$, where B is a small discrete disk-like set. (c) Peaks: $f - f \circ B$. (d) Valleys: $f \bullet B - f$.



(a) (b)

Figure 4: (a) Original signal (dashed line) and its blurring (solid line) via convolution with a finite positive symmetric tapered impulse response. (b) Filtered versions of the blurred signal in (a) produced by iterating the 1D toggle filter (with $B = \{-1, 0, 1\}$) until convergence to the limit signal reached at 125 iterations; the displayed filtered signals correspond to iteration indexes that are multiples of 20.



(a) (b)

(c) (d)

Figure 5: (a) Original image f. (b) Blurred image g obtained by an out-of-focus camera digitizing f. (c) Output of the 2D toggle filter acting on g (B was a small symmetric disk-like set containing the origin). (d) Limit of iterations of the toggle filter on g (reached at 150 iterations).

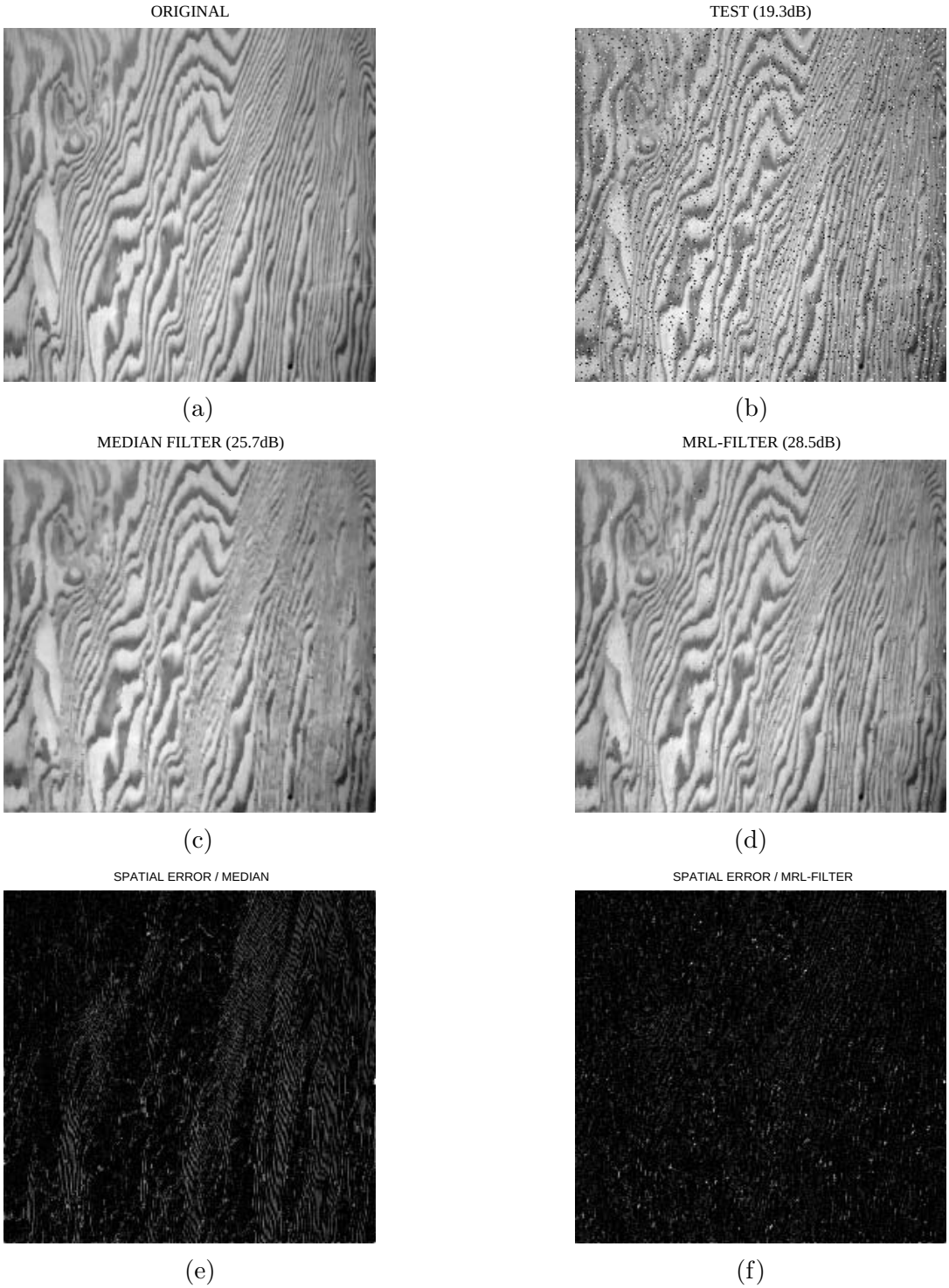


Figure 6: (a) Original clean texture image (240x250). (b) Noisy image: Image (a) corrupted by a hybrid 47dB additive Gaussian white noise and 10% multi-valued impulse noise (PSNR = 19.3dB). (c) Noisy image restored by a flat 3x3 median filter (PSNR = 25.7dB). (d) Noisy image restored by the designed 3x3 MRL-filter (PSNR = 28.5dB). (e) Spatial error map of the flat median filter; lighter areas indicate higher errors. (f) Spatial error map of the MRL-filter.