Chapter 3.3 for the Book:
*The Image and Video Processing Handbook, 2nd edition*, Edited by A. C. Bovik, Elsevier Academic Press, 2005, pp.135-156.

# MORPHOLOGICAL FILTERING FOR IMAGE ENHANCEMENT AND FEATURE DETECTION

Petros Maragos

National Technical University of Athens
School of Electrical & Computer Engineering
Zografou 15773, Athens, Greece.
Email: maragos@cs.ntua.gr

## 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. Image enhancement is usually followed by (or is done simultaneously with) *detection of features* such as edges, peaks, and other geometric features which is of paramount importance in low-level vision. Further, many related vision problems involve the detection of a known template; such problems are usually solved via template matching.

While traditional approaches for solving the above tasks have used mainly tools of linear systems, nowadays a new understanding has matured 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 geometric 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 [13, 24] 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 [19, 18] on binary pattern recognition based on cellular automata and Boolean/threshold logic. In the late 1970's it was extended to gray-level images [24]. In the mid 1980's it was brought to the mainstream of image/signal processing and related to other nonlinear filtering approaches [10, 11]. Finally, in the late 1980's and 1990's it was generalized to arbitrary lattices [25, 6]. 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 feature detection. Thus, it discusses four important general problems of low-level (early) vision, progressing from the easiest (or more easily defined) to the more difficult (or harder to define): (i) cleaning the image from noise or improving its contrast; (ii) detecting in the image the presence of known templates; (iii) detecting the existence and location of geometric features such as edges and peaks whose types are known but not their exact form; (iv) designing optimal morphological filters.

# 2 Morphological Image Operators

## 2.1 Morphological Filters for Binary Images

Given a sampled[1] 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 [19, 18].

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 $B$:

$$X \oplus B \triangleq \{z : (B^s)_{+z} \cap X \neq \emptyset\} = \bigcup_{y \in B} X_{+y} \tag{2}$$

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

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

[1] Signals of a continuous variable $x \in \mathbb{R}^m$ are usually denoted by $f(x)$, whereas for signals with discrete variable $x \in \mathbb{Z}^m$ 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 Minkowski **opening** $X \circ B \triangleq (X \ominus B) \oplus B$ and the **closing** $X \bullet B \triangleq (X \oplus B) \ominus B$ 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 $X \oplus B = (X^c \ominus B^s)^c$; i.e., dilation of an image object by $B$ is equivalent to eroding its background by $B^s$ 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 $\mathbb{E} = \mathbb{R}^2$ 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 upper **level sets** (also called *threshold sets*)

$$X_v(f) \triangleq \{x \in \mathbb{E} : f(x) \geq v\} \ , \quad -\infty < v < +\infty \tag{4}$$

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

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

where 'sup' denotes supremum[2]. Transforming each level set of the input signal $f$ by a set operator $\Psi$ and viewing the transformed sets as level sets of a new image creates [24, 10] a **flat** image operator $\psi$ whose output signal is

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

For example, if $\Psi$ 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 $B$:

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

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

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.

[2] 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:

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

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

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[3]

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 [10, 11] 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 $\Psi$ is uniquely characterized by its **kernel** $\text{Ker}(\Psi) \triangleq \{X \in \mathbb{Z}^2 : 0 \in \Psi(X)\}$. If $\Psi$ is also *increasing* [i.e., $X \subseteq Y \Longrightarrow \Psi(X) \subseteq \Psi(Y)$], then it can be represented as a union of erosions by all its kernel sets [13]. However, this kernel representation requires an infinite number of erosions. A more efficient (requiring less erosions) representation uses only a substructure of the kernel, its **basis** $\text{Bas}(\Psi)$, defined as the collection of kernel elements that are *minimal* with respect to the partial ordering $\subseteq$. If $\Psi$ is also *upper semicontinuous* [i.e., $\Psi(\bigcap_n X_n) = \bigcap_n \Psi(X_n)$ for any decreasing set sequence $(X_n)$], then $\Psi$ has a nonempty basis and can be represented exactly as a union of erosions by its basis sets:

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

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

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

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

[3] 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 $(W^s)_{+x}$ in decreasing order and picking the $p$-th largest value, $p = 1, 2, ..., n$, yields the output signal from the $p$-th **rank filter**:

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

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.

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

$$S \Box_p W \triangleq \{x : \mathrm{card}[(W^s)_{+x} \cap S] \geq p\} \tag{14}$$

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

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

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

where $X_v(f)$ is the level set (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 $\psi$ that have a corresponding set operator $\Psi$ 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 $X_v(f)$ 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 $\Psi$ in (6) creates a class of nonlinear signal operators via threshold superposition, called **stack filters** [2, 10]

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

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) [10]. 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 [25, 6] 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 $\mathbb{E}$ where the $\bigvee / \bigwedge$ lattice operations are the set union/intersection, and (ii) the space of all graylevel image signals $f : \mathbb{E} \to \overline{\mathbb{R}}$ where the $\bigvee / \bigwedge$ lattice operations are the supremum/infimum of sets of real numbers. An operator $\psi$ 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:

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

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

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

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

where $I$ is an arbitrary index set, *idempotence* of an operator $\psi$ means that $\psi(\psi(f)) = \psi(f)$, and *anti-extensivity* and *extensivity* of operators $\alpha$ and $\beta$ 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 Minkowski dilations $\oplus$, erosions $\ominus$, openings $\circ$, and closings $\bullet$ are simple special cases of their lattice counterparts.

In lattice-theoretic morphology, the term *morphological filter* means any increasing and idempotent operator on a lattice of images. However, in this chapter we shall use the term 'morphological operator', which broadly means a morphological signal transformation, interchangeably with the term 'morphological filter', in analogy to the terminology 'rank or linear filter'.

## 3 Morphological Filters for Image Enhancement

Enhancement may be accomplished in various ways including (i) noise suppression, (ii) simplification by retaining only those image components that satisfy certain size or contrast criteria, (iii) contrast sharpening. The first two cases may also be viewed as examples of 'image smoothing'.

The simplest morphological image smoother is a Minkowski opening by a disk $B$. This smooths and simplifies a (binary image) set $X$ by retaining only those parts inside which a translate of $B$ can fit. Namely,

$$X \circ B = \bigcup_{B_{+z} \subseteq X} B_{+z} \tag{21}$$

In case of graylevel image $f$, its opening by $B$ performs the above smoothing at all level sets simultaneously. However, this horizontal geometric local and isotropic smoothing performed by the Minkowski disk opening may not be sufficient for several other smoothing tasks that may need directional smoothing, or may need contour preservation based on size or contrast criteria. To deal with these issues, we discuss below several types of morphological filters that are generalized operators in the lattice-theoretic sense and have proven to be very useful for image enhancement.

## 3.1 Noise Suppresion and Image Smoothing

### 3.1.1 Median versus Opening-Closing

In their behavior as nonlinear smoothers, as shown in Fig. 1, 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 [10]. 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.

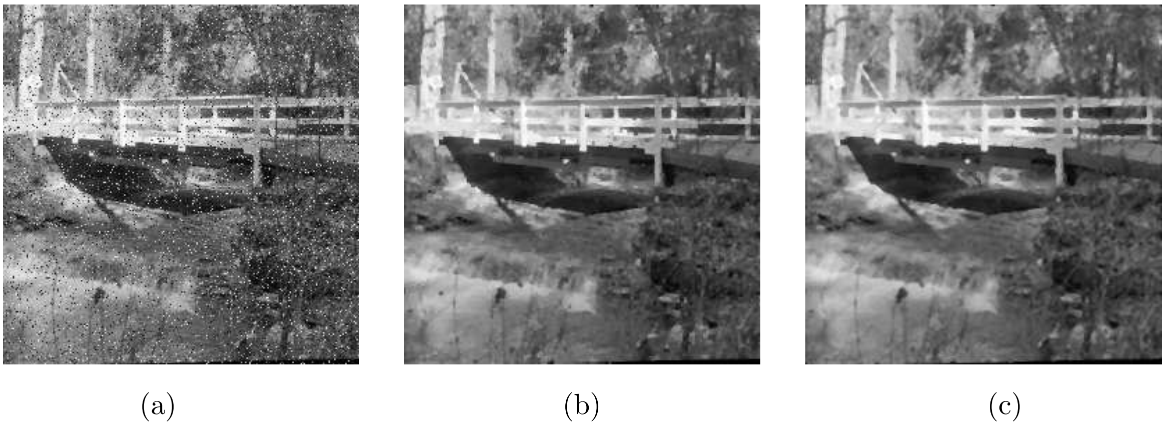


Figure 1: (a) Noisy image obtained by corrupting an original with two-level salt-and-pepper noise occuring with probability 0.1 (PSNR=18.9 dB). (b) Open-closing of noisy image by a $2 \times 2$-pel square (PSNR=25.4 dB). (c) Median of noisy image by a $3 \times 3$-pel square (PSNR=25.4 dB).

The popularity and efficiency of the simple morphological openings and closings to suppress impulse noise is supported by the following theoretical development [27]. 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 [13, 24] 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 [27] of the original $X$ given the noisy image $Y$ is the opening (or closing) of the observed $Y$ by $B$.

### 3.1.2 Alternating Sequential Filters

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

$$\psi_{asf}(f) = \beta_r \alpha_r ... \beta_2 \alpha_2 \beta_1 \alpha_1(f) \tag{22}$$

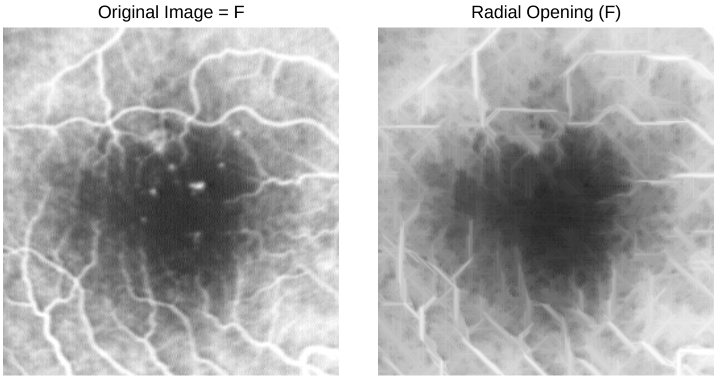
called **alternating sequential filters (ASF)**, which smooth progressively from the smallest scale possible up to a maximum scale $r$ and have a broad range of applications [25]. Their optimal design is addressed in [23]. Further, the Minkowski open-closings in an ASF can be replaced by other types of lattice open-closings. A simple such generalization is the radial open-closing, discussed next.

### 3.1.3 Radial Openings

Consider a 2D image $f$ that contains 1D objects, e.g. lines; then the simple Minkowski opening or closing of $f$ by a disk $B$ 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**

$$\alpha(f) = \bigvee_{\theta} f \circ L_\theta \tag{23}$$

where the sets $L_\theta$ 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 $L_\theta$ in at least one of the possible orientations $\theta$. See Fig. 2 for an example. Dually, in case of dark 1D objects, we can use a radial closing $\beta(f) = \bigwedge_\theta f \bullet L_\theta$.



(a) (b) (c)

Figure 2: (a) Original image $f$ of an eye angiogram with microaneurisms. (b) Radial opening $\alpha(f)$ of $f$ as max of four openings by lines oriented at $0^o, 45^o, 90^o, 135^o$ of size 20 pixels each. (c) Reconstruction opening $\rho^-(\alpha(f)|f)$ of $f$ using the radial opening as marker.

## 3.2 Connected Filters for Smoothing and Simplification

The **flat zones** of an image signal $f : \mathbb{E} \to \overline{\mathbb{R}}$ are defined as the connected components of the image domain on which $f$ assumes a constant value. A useful class of morphological filters was introduced in [26, 22] which operate by merging flat zones and hence exactly preserving the contours of the image parts remaining in the filter's output. These are called **connected operators**. They cannot create new image structures or new boundaries if they did not exist in the input. Specifically, if $\mathcal{D}$ is a partition of the image domain, let $\mathcal{D}(x)$ denote the (partition member) region that contains the pixel $x$. Now, given two partitions $\mathcal{D}_1, \mathcal{D}_2$, we say that $\mathcal{D}_1$ is 'finer' than $\mathcal{D}_2$ if $\mathcal{D}_1(x) \subseteq \mathcal{D}_2(x)$ for all $x$. An operator $\psi$ is called *connected* if the flat zone partition of its input $f$ is finer than the flat zone partition of its output $\psi(f)$. Next we discuss two types of connected operators, the area filters and the reconstruction filters.

### 3.2.1 Area Openings

There are numerous image enhancement problems where what is needed is suppression 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** $\alpha_n$ of size $n$ which, for binary images, keeps only the connected components whose area is $\geq n$ and eliminates the rest. Consider an input set $X = \bigsqcup_i X_i$ as a union of disjoint connected components $X_i$. Then the output from the area opening is

$$\alpha_n(X) = \bigsqcup_{\text{Area}(X_j) \geq n} X_j, \quad X = \bigsqcup_i X_i \tag{24}$$

where $\bigsqcup$ denotes disjoint union. The area opening can be extended to graylevel images $f$ by applying the same binary area opening to all level sets of $f$ and constructing the filtered graylevel image via threshold superposition:

$$\alpha_n(f)(x) = \sup\{v : x \in \alpha_n[X_v(f)]\} \tag{25}$$

Figure 3 shows examples of binary and gray area openings. If we apply the above operations to the complements of the level sets of an image, we obtain an *area closing.*

### 3.2.2 Reconstruction Filters and Levelings

Consider a *reference* (image) set $X = \bigsqcup_i X_i$ as a union of $I$ disjoint connected components $X_i$, $i \in I$, and let $M \subseteq X_j$ be a *marker* in some component(s) $X_j$, indexed by $j \in J \subseteq I$; i.e., $M$ could consist of a single point or some feature sets in $X$ that lie only in the component(s) $X_j$. Let us define the **reconstruction opening** as the operator

$$\rho^-(M|X) \triangleq \text{connected components of } X \text{ intersecting } M. \tag{26}$$

Its output contains exactly the input component(s) $X_j$ that intersect the marker. It can extract large-scale components of the image from knowledge only of a smaller marker inside them. Note that the reconstruction opening has two inputs. If the marker $M$ is fixed, then the mapping $X \mapsto \rho^-(M|X)$ is a lattice opening since it is increasing, anti-extensive and idempotent. Its output is called the *morphological reconstruction* of (the component(s) of) $X$ from the marker $M$. However, if the reference $X$ is fixed, then the mapping $M \mapsto \rho^-(M|X)$ is an idempotent lattice dilation; in this case, the output is called the reconstruction of $M$ under $X$.

An algorithm to implement the discrete reconstruction opening is based on the *conditional dilation* of $M$ by $B$ within $X$:

$$\delta_B(M|X) \triangleq (M \oplus B) \cap X \tag{27}$$

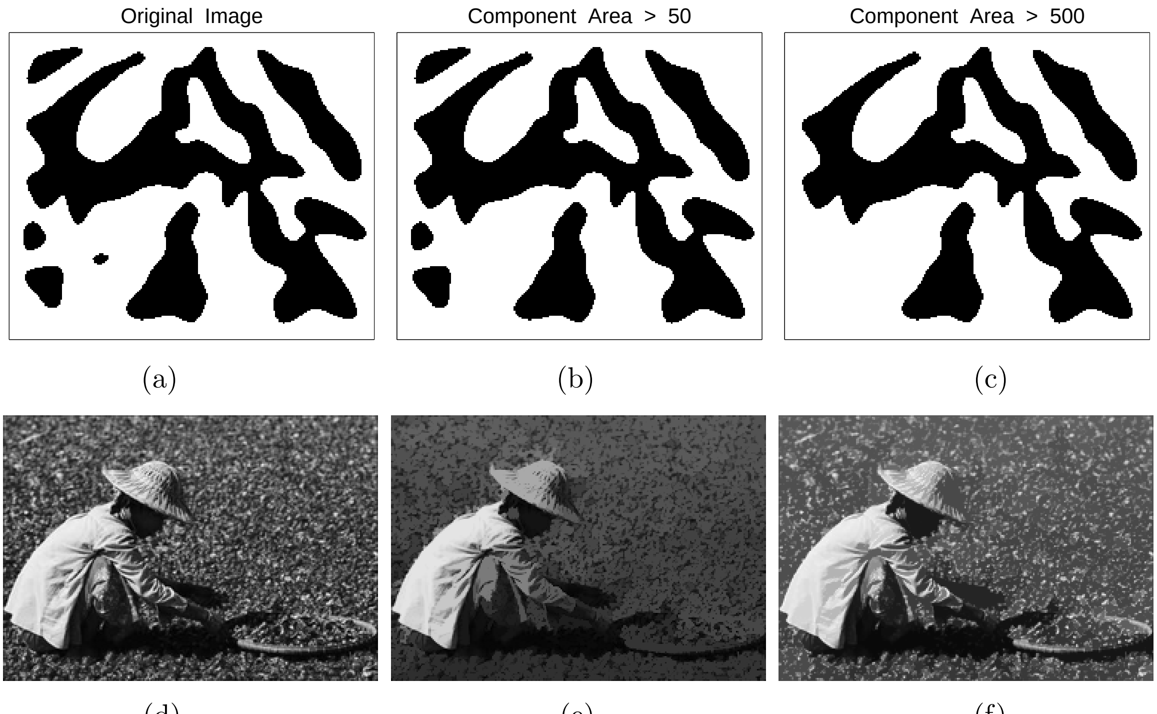


Figure 3: Top row: (a) Original binary image ($192 \times 228$ pixels). (b) Area opening by keeping connected components with area $\geq 50$. (c) Area opening by keeping components with area $\geq 500$. Bottom row: (d) Gray original image ($420 \times 300$ pixels). (e) Gray area opening by keeping bright components with area $\geq 500$. (f) Gray area closing by keeping dark components with area $\geq 500$.

where $B$ is the unit-radius discrete disk associated with the selected connectivity of the rectangular grid; i.e., a 5-pixel rhombus or a 9-pixel square depending on whether we have 4- or 8-neighbor connectivity, respectively. By iterating this conditional dilation we can obtain in the limit the whole marked component(s) $X_j$, i.e. the *conditional* reconstruction opening

$$\rho_B^-(M|X) = \lim_{k\to\infty} Y_k, \quad Y_k = \delta_B(Y_{k-1}|X), \ \ Y_0 = M. \tag{28}$$

An example is shown in Fig. 4.

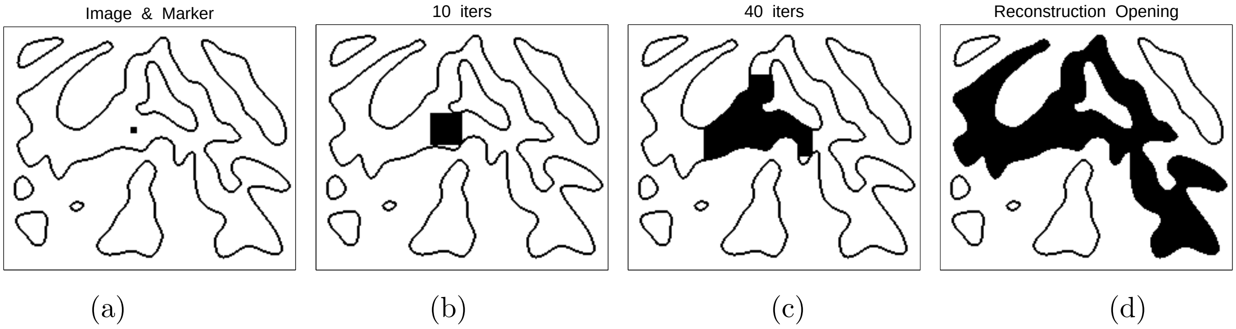


Figure 4: (a) Original binary image ($192 \times 228$ pixels) and a square marker within the largest component. The next three images show iterations of the conditional dilation of the marker with a $3 \times 3$-pixel square structuring element. (b) 10 iterations. (c) 40 iterations. (d) Reconstruction opening, reached after 128 iterations.

Replacing the binary with graylevel images, the set dilation with function dilation, and $\cap$ with

$\wedge$ yields the graylevel reconstruction opening of a graylevel image $f$ from a marker image $m$:

$$\rho_B^-(m|f) = \lim_{k\to\infty} g_k, \quad g_k = \delta_B(g_{k-1}) \wedge f, \;\; g_0 = m \leq f. \tag{29}$$

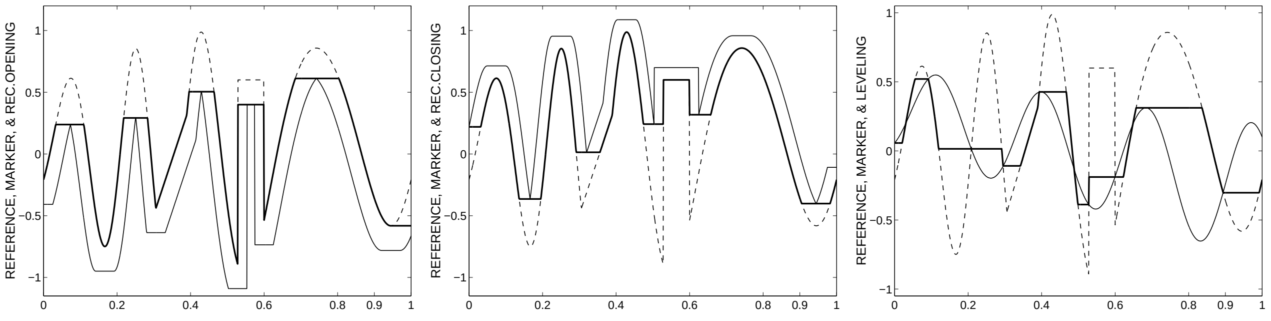
This reconstructs the bright components of the reference image $f$ that contain the marker $m$. For example, as shown in Fig. 2, the results of any prior image smoothing, like the radial opening of Fig. 2(b), can be treated as a marker which is subsequently reconstructed under the original image as reference to recover exactly those bright image components whose parts have remained after the first operation.

There is a large variety of reconstruction openings depending on the choise of the marker. Two useful cases are (i) *size*-based markers chosen as the Minkowski erosion $m = f \ominus rB$ of the reference image $f$ by a disk of radius $r$, and (ii) *contrast*-based markers chosen as the difference $m(x) = f(x) - h$ of a constant $h > 0$ from the image. In the first case the reconstruction opening retains only objects whose horizontal size (i.e. diameter of inscribable disk) is not smaller than $r$. In the second case only objects whose contrast (i.e. height difference from neighbors) exceeds $h$ will leave a remnant after the reconstruction. In both cases, the marker is a function of the reference signal.

Reconstruction of the dark image components hit by some marker is accomplished by the dual filter, the *reconstruction closing*

$$\rho_B^+(m|f) = \lim_{k\to\infty} g_k, \quad g_k = \varepsilon_B(g_{k-1}) \vee f, \;\; g_0 = m \geq f. \tag{30}$$

Examples of graylevel reconstruction filters are shown in Fig. 5.



(a) (b) (c)

Figure 5: Reconstruction filters for 1D images. Each figure shows reference signals $f$ (dash), markers (thin solid), and reconstructions (thick solid). (a) Reconstruction opening from marker$=(f \ominus B) - \text{const.}$ (b) Reconstruction closing from marker$=(f \oplus B) + \text{const.}$ (c) Leveling (self-dual reconstruction) from an arbitrary marker.

Despite their many applications, reconstruction openings and closings $\psi$ have as disadvantage the property that they are *not* self-dual operators; hence, they treat the image and its background asymmetrically. A newer operator type that unifies both of them and possesses self-duality is the leveling [15]. Levelings are nonlinear object-oriented filters that simplify a reference image $f$ through a simultaneous use of locally expanding and shrinking an initial seed image, called the marker $m$, and globally constraining of the marker evolution by the reference image. Specifically, iterations of the image operator $\lambda(m|f) = (\delta_B(m) \wedge f) \vee \varepsilon_B(m)$, where $\delta_B(\cdot)$ (resp. $\varepsilon_B(\cdot)$) is a dilation (resp. erosion) by the unit-radius discrete disk $B$ of the grid, yield in the limit the **leveling** of $f$ w.r.t. $m$:

$$\Lambda_B(m|f) = \lim_{k\to\infty} g_k, \quad g_k = (\delta_B(g_{k-1}) \wedge f) \vee \varepsilon_B(g_{k-1}), \quad g_0 = m \tag{31}$$

In contrast to the reconstruction opening (closing) where the marker $m$ is smaller (greater) than $f$, the marker for a general leveling may have an arbitrary ordering w.r.t. the reference signal. See Fig. 5(c) for an example. The leveling reduces to being a reconstruction opening (closing) over regions where the marker is smaller (greater) than the reference image.

If the marker is self-dual, then the leveling is a self-dual filter and hence treats symmetrically the bright and dark objects in the image. Thus, the leveling may be called a *self-dual reconstruction* filter. It simplifies both the original image and its background by completely eliminating smaller objects inside which the marker cannot fit. The reference image plays the role of a *global constraint*.

In general, levelings have many interesting multiscale properties [15]. For example, they preserve the coupling and sense of variation in neighbor image values and do not create any new regional maxima or minima. Also, they are increasing and idempotent filters. They have proven to be very useful for image simplification toward segmentation because they can suppress small-scale noise or small features and keep only large-scale objects with exact preservation of their boundaries.

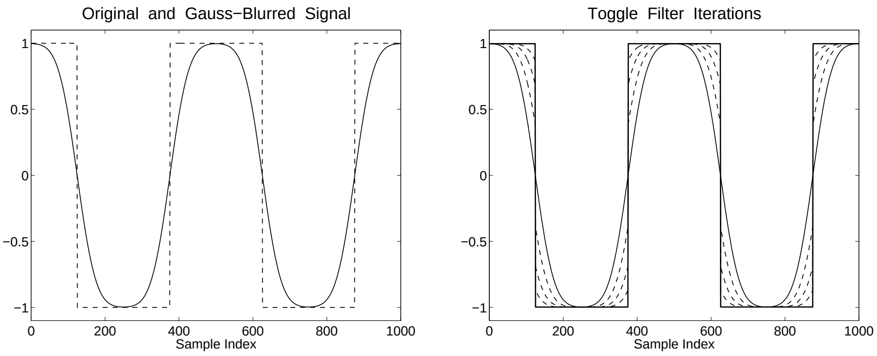
## 3.3 Contrast Enhancement

Imagine a graylevel image $f$ that has resulted from blurring an original image $g$ by linearly convolving it with a Gaussian function of variance $2t$. This Gaussian blurring can be modeled by running the classic heat diffusion differential equation for the time interval $[0, t]$ starting from the initial condition $g$ at $t = 0$. If we can reverse in time this diffusion process, then we can deblur and sharpen the blurred image. By approximating the spatio-temporal derivatives of the heat equation with differences, we can derive a linear discrete filter that can enhance the contrast of the blurred image $f$ by subtracting from $f$ a discretized version of its Laplacian $\nabla^2 f = \partial^2 f/\partial x^2 + \partial^2 f/\partial y^2$. This is a simple linear deblurring scheme, called *unsharp constrast enhancement*. A conceptually similar procedure is the following nonlinear filtering scheme.

Consider a graylevel image $f[x]$ and a small-size symmetric disk-like structuring element $B$ containing the origin. The following discrete nonlinear filter [7] can enhance the local contrast of $f$ by sharpening its edges:

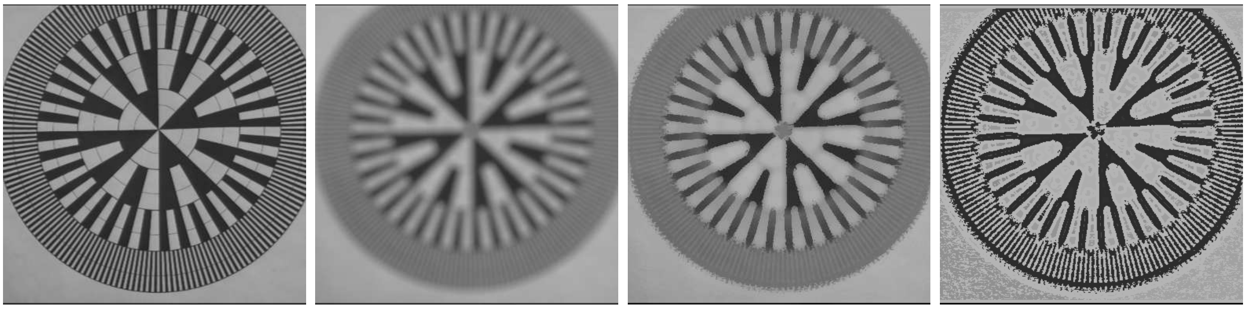
$$\psi(f)[x] = \begin{cases} (f \oplus B)[x] & \text{if} \quad f[x] \geq ((f \oplus B)[x] + (f \ominus B)[x])/2 \\ (f \ominus B)[x] & \text{if} \quad f[x] < ((f \oplus B)[x] + (f \ominus B)[x])/2 \end{cases} \tag{32}$$

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)* [7] reached after a finite number of iterations. Examples are shown in Fig. 6 and Fig. 7.



(a) (b)

Figure 6: (a) Original signal (dashed line) $f[x] = \mathrm{sign}(\cos(4\pi x))$, $x \in [0,1]$, and its blurring (solid line) via convolution with a truncated sampled Gaussian function of $\sigma = 40$. (b) Filtered versions (dashed lines) of the blurred signal in (a) produced by iterating the 1D toggle filter (with $B = \{-1,0,1\}$) until convergence to the limit signal (thick solid line) reached at 66 iterations; the displayed filtered signals correspond to iteration indexes that are multiples of 20.



(a) (b) (c) (d)

Figure 7: (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). (d) Limit of iterations of the toggle filter on $g$ (reached at 150 iterations).

# 4 Morphological Operators for Template Matching

## 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* $L_{fg}[y] \triangleq \sum_{x \in W} f[x+y]g[x]$ 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 [11] 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

$$M_{fg}[y] \triangleq \sum_{x \in W} \min(f[x+y], g[x]) \tag{33}$$

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:

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

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., $G = G^s$.) 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*

$$\frac{Pr(f/H_1)}{Pr(f/H_0)} \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} \frac{Pr(H_0)}{Pr(H_1)} \tag{34}$$

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

$$M_{fg}[y] = \sum_{x \in W} \min(f[x], g[x-y]) \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} \theta = \frac{1}{2}\left( \frac{\log[Pr(H_0)/Pr(H_1)]}{\log[(1-p)/p]} + \mathrm{card}(G) \right) \tag{35}$$

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 $\theta$. This is equivalent[4] to performing a $r$-th rank filtering on $f$ by a set $G$ equal to the support of $g$, where $1 \leq r \leq \mathrm{card}(G)$ and $r$ is related to $\theta$. 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 $r = \theta = \mathrm{card}(G)/2$ 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 $(A, B)$:

$$X \otimes (A, B) \triangleq \{x : A_{+x} \subseteq X,\ B_{+x} \subseteq X^c\} \tag{36}$$

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 [24]. It can actually model all binary template matching schemes in binary pattern recognition that use a pair of a positive and a negative template [19].

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.

## 5 Morphological Operators for Feature Detection

### 5.1 Edge Detection

By image *edges* we define *abrupt intensity changes* of an image. Intensity changes usually correspond to physical changes in some property of the imaged 3D objects' surfaces (e.g. changes in reflectance, texture, depth or orientation discontinuities, object boundaries) or changes in their illumination. Thus, edge detection is very important for subsequent higher-level vision tasks and can lead to some inference about physical properties of the 3D world. Edge types may be classified into three types by approximating their shape with three idealized patterns: lines, steps, and roofs, which correspond, respectively, to the existence of a Dirac impulse in the derivative of order 0, 1, and 2. Next we focus mainly on *step* edges. The problem of edge detection can be separated into three main subproblems:

[4] 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 $n = \mathrm{card}(G)$ 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.

*1. Smoothing*: image intensities are smoothed via filtering or approximated by smooth analytic functions. The main motivations are to suppress noise and decompose edges at multiple scales.
*2. Differentiation*, which amplifies the edges and creates more easily detectable simple geometric patterns.
*3. Decision*: edges are detected as peaks in the magnitude of the first-order derivatives or zerocrossings in the second-order derivatives, both compared with some threshold.

Smoothing can be either linear or nonlinear. Similarly for the differentiation. Further the differentiation can be either directional or isotropic. Next, after a brief synopsis of the main linear approaches for edge detection, we describe some fully nonlinear ones using morphological gradient-type residuals.

### 5.1.1 Linear Edge Operators

In linear edge detection, both smoothing and differentiation are done via linear convolutions. These two stages of smoothing and differentiation can be done in a single stage of convolution with the derivative of the smoothing kernel. Three well-known approaches for edge detection using linear operators in the main stages are the following:

- **Convolution with Edge Templates:** Historically, the first approach for edge detection, which lasted for about three decades (1950s-1970s), was to use discrete approximations to the image linear partial derivatives $f_x = \partial f/\partial x$ and $f_y = \partial f/\partial y$ by convolving the digital image $f$ with very small edge-enhancing kernels. Examples include the Prewitt, Sobel and Kirsch edge convolution masks reviewed in [19, 5]. Then these approximations to $f_x, f_y$ were combined nonlinearly to give a gradient magnitude $||\nabla f||$ using the $\ell_1$, $\ell_2$ or $\ell_\infty$ norm. Finally, peaks in this edge gradient magnitude were detected, via thresholding, for a binary edge decision. Alternatively, edges were identified as zerocrossings in second-order derivatives which were approximated by small convolution masks acting as digital Laplacians. All these above approaches do not perform well because the resulting convolution masks act as poor digital highpass filters that amplify high-frequency noise and do not provide a scale localization/selection.

- **ZeroCrossings of Laplacian-of-Gaussian Convolution:** Marr & Hildreth [12] developed a theory of edge detection based on evidences from biological vision systems and ideas from signal theory. For image smoothing they chose linear convolutions with isotropic Gaussian functions $G_\sigma(x, y) = \exp[-(x^2 + y^2)/2\sigma^2]/(2\pi\sigma^2)$ to optimally localize edges both in space and frequency domains. For differentiation they chose the Laplacian operator $\nabla^2$, since it is the only isotropic linear second-order differential operator. The combination of Gaussian smoothing and Laplacian can be done using a single convolution with a Laplacian-of-Gaussian (LoG) kernel, which is an approximate bandpass filter that isolates from the original image a scale band on which edges are detected. The scale is determined by $\sigma$. Thus, the image edges are defined as the zerocrossings of the image convolution with a LoG kernel. In practice, one does not accept all zerocrossings in the LoG output as edge points but tests whether the slope of the LoG output exceeds a certain threshold.

- **ZeroCrossings of Directional Derivatives of Smoothed Image:** For detecting edges in 1D signals corrupted by noise, Canny [3] developed an optimal approach where edges were detected as maxima in the output of a linear convolution of the signal with a finite-extent impulse response $h$. By maximizing the following figures of merit, (i) good detection in terms of robustness to noise, (ii) good edge localization, and (iii) uniqueness of the result in the vicinity of the edge, he found an optimum filter with an impulse response $h(x)$ which can be closely approximated by the derivative of a Gaussian. For 2D images, the Canny edge detector consists of three steps: (1) Smooth the image $f(x, y)$ with an isotropic 2D Gaussian $G_\sigma$. (2) Find the zerocrossings of the second-order directional derivative $\partial^2 f/\partial\eta^2$ of the image in the direction of the gradient $\vec{\eta} = \nabla f/||\nabla f||$. (3) Keep only those zerocrossings and declare them as edge pixels if they belong to connected arcs whose points possess edge strengths that pass a double-threshold hysteresis criterion. Closely related to Canny's edge detector was Haralick's previous work (reviewed in [5]) to regularize the 2D discrete image function by fitting to it bicubic interpolating polynomials, compute the image derivatives from the interpolating polynomial, and find the edges as the zerocrossings of the second directional derivative in the gradient direction. The Haralick-Canny edge detector yields different and usually better edges than the Marr-Hildreth detector.

### 5.1.2 Morphological Edge Detection

The **boundary** of a set $X \subseteq \mathbb{R}^m$, $m = 1, 2, ...$, is given by

$$\partial X \triangleq \overline{X} \setminus \overset{\circ}{X} = \overline{X} \cap (\overset{\circ}{X})^c \tag{37}$$

where $\overline{X}$ and $\overset{\circ}{X}$ denote the closure and interior of $X$. Now, if $||x||$ is the Euclidean norm of $x \in \mathbb{R}^m$, $B$ is the unit ball, and $rB = \{x \in \mathbb{R}^m : ||x|| \leq r\}$ is the ball of radius $r$, then it can be shown that

$$\partial X = \bigcap_{r>0} (X \oplus rB) \setminus (X \ominus rB) \tag{38}$$

Hence, the set difference between erosion and dilation can provide the 'edge', i.e., the boundary of a set $X$.

These ideas can also be extended to signals. Specifically, let us define morphological **sup-derivative** $\mathcal{M}(f)$ of a function $f : \mathbb{R}^m \to \mathbb{R}$ at a point $x$ as

$$\mathcal{M}(f)(x) \triangleq \lim_{r \downarrow 0} \frac{(f \oplus rB)(x) - f(x)}{r} = \lim_{r \downarrow 0} \frac{\bigvee_{||y|| \leq r} f(x+y) - f(x)}{r} \tag{39}$$

By applying $\mathcal{M}$ to $-f$ and using the duality between dilation and erosion, we obtain the *inf-derivative* of $f$. Suppose now that $f$ is differentiable at $x = (x_1, ..., x_m)$ and let its gradient be $\nabla f = \left(\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_m}\right)$. Then, it can be shown that

$$\mathcal{M}(f)(x) = ||\nabla f(x)|| \tag{40}$$

Next, if we take the difference between sup-derivative and inf-derivative when the scale goes to zero, we arrive at an isotropic *second-order morphological derivative*:

$$\mathcal{M}^2(f)(x) \triangleq \lim_{r \downarrow 0} \frac{[(f \oplus rB)(x) - f(x)] - [f(x) - (f \ominus rB)(x)]}{r^2} \tag{41}$$

The peak in the first-order morphological derivative or the zerocrossing in the second-order morphological derivative can detect the location of an edge, in a similar way as the traditional linear derivatives can detect an edge.

By approximating the morphological derivatives with differences, various simple and effective schemes can be developed for extracting edges in *digital images*. For example, for a *binary* discrete image represented as a set $X$ in $\mathbb{Z}^2$, the set difference $(X \oplus B) \setminus (X \ominus B)$ gives the boundary of $X$. Here $B$ equals the 5-pixel rhombus or 9-pixel square depending on whether we desire 8- or 4-connected image boundaries. An asymmetric treatment between the image foreground and background results if the dilation difference $(X \oplus B) \setminus X$ or the erosion difference $X \setminus (X \ominus B)$ is applied, because they yield a boundary belonging only to $X^c$ or to $X$, respectively.

Similar ideas apply to graylevel images. Both the *dilation residual* and the *erosion residual*

$$\text{edge}^{\oplus}(f) \triangleq (f \oplus B) - f, \quad \text{edge}^{\ominus}(f) \triangleq f - (f \ominus B) \tag{42}$$

enhance the edges of a graylevel image $f$. Adding these two operators yields the **discrete morphological gradient**

$$\text{edge}(f) \triangleq (f \oplus B) - (f \ominus B) = \text{edge}^{\oplus}(f) + \text{edge}^{\ominus}(f) \tag{43}$$

that treats more symmetrically the image and its background; see Fig. 8 for examples.

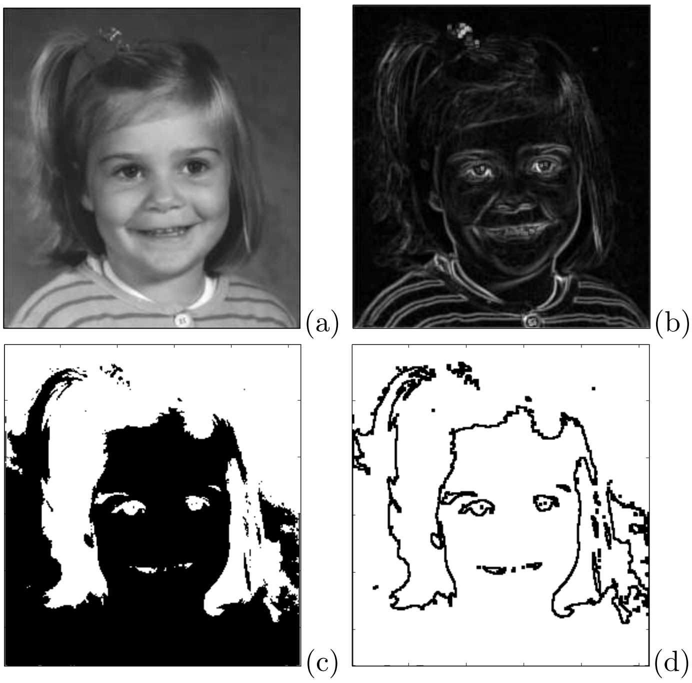


Figure 8: (a) Original image $f$ with range in $[0, 255]$. (b) $f \oplus B - f \ominus B$, where $B$ is a $3 \times 3$-pixel square. (c) Level set $X = X_i(f)$ of $f$ at level $i = 100$. (d) $X \oplus B \setminus X \ominus B$. (In (c)-(d) black areas represent the sets, while white areas are the complements.)

Threshold analysis can be used to understand the action of the above edge operators. Let the non-negative discrete-valued image signal $f(x)$ have $L + 1$ possible integer intensity values: $i = 0, 1, ..., L$. By thresholding $f$ at all levels we obtain the *threshold binary images* $f_i$ from which we can resynthesize $f$ via threshold-sum signal superposition:

$$f(x) = \sum_{i=1}^{L} f_i(x), \quad f_i(x) = \begin{cases} 1, & \text{if} \quad f(x) \geq i \\ 0, & \text{if} \quad f(x) < i \end{cases} \tag{44}$$

Since the flat dilation and erosion by a finite $B$ commute with thresholding and $f$ is nonnegative, they obey threshold-sum superposition. Therefore, the dilation-erosion difference operator also obeys threshold-sum superposition:

$$\text{edge}(f) = \sum_{i=1}^{L} \text{edge}(f_i) = \sum_{i=1}^{m} f_i \oplus B - f_i \ominus B \tag{45}$$

This implies that the output of the edge operator acting on the graylevel image $f$ is equal to the sum of the binary signals that are the boundaries of the binary images $f$. See Fig. 8. At each pixel $x$, the larger the gradient of $f$, the larger the number of threshold levels $i$ such that $\text{edge}(f_i)(x) = 1$, and hence the larger the value of the graylevel signal $\text{edge}(f)(x)$. Finally, a binarized edge image can be obtained by thresholding $\text{edge}(f)$ or detecting its peaks.

The morphological digital edge operators have been extensively applied to image processing by many researchers. By combining the erosion and dilation differences, various other effective edge operators have also been developed. Examples include 1) the *asymmetric morphological edge-strength operators* by Lee et al. [8]

$$\min[\text{edge}^{\ominus}(f), \text{edge}^{\oplus}(f)], \quad \max[\text{edge}^{\ominus}(f), \text{edge}^{\oplus}(f)] \tag{46}$$

and 2) the edge operator $\text{edge}^{\oplus}(f) - \text{edge}^{\ominus}(f)$ by Vliet et al. [29], which behaves as a discrete *"nonlinear Laplacian"*

$$NL(f) = (f \oplus B) + (f \ominus B) - 2f \tag{47}$$

and its zero-crossings can yield edge locations. Actually, for a 1D twice differentiable function $f(x)$ it can be shown that, if $df(x)/dx \neq 0$, then $\mathcal{M}^2(f)(x) = d^2 f(x)/dx^2$.

For robustness in the presence of noise these morphological edge operators should be applied after the input image has been smoothed first either via linear or nonlinear filtering. For example, in [8] a small local averaging is used on $f$ before applying the morphological edge-strength operator, resulting in the so-called *min-blur* edge detection operator

$$\min[f_{av} - f_{av} \ominus B, f_{av} \oplus B - f_{av}] \tag{48}$$

with $f_{av}$ being the local average of $f$, whereas in [4] an opening and closing is used instead of linear pre-averaging:

$$\min[f \circ B - f \ominus B, f \oplus B - f \bullet B] \tag{49}$$

Combinations of such smoothings and morphological first or second derivatives have performed better in detecting edges of noisy images. See Fig. 9 for an experimental comparison of the LoG and the morphological second derivative in detecting edges.

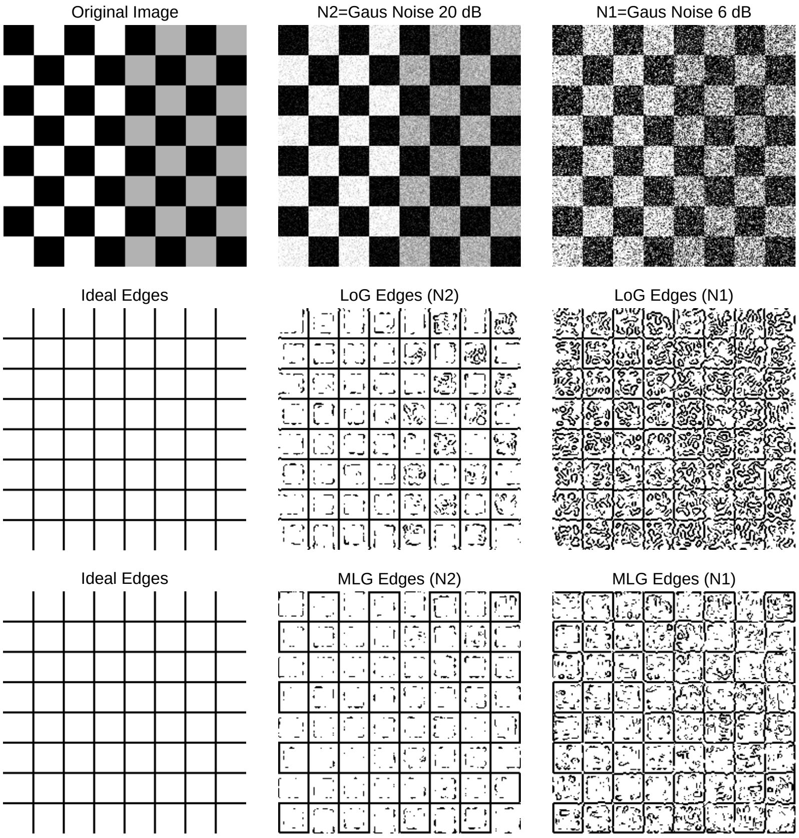


Figure 9: Top: Test image and two noisy versions with additive Gaussian noise at SNR 20 dB and 6 dB. Middle: Ideal edges, and edges from zerocrossings of Laplacian-of-Gaussian of the two noisy images. Bottom: Ideal edges, and edges from zerocrossings of 2D morphological second derivative (nonlinear Laplacian) of the two noisy images after some Gaussian presmoothing. In both methods, the edge pixels were the subset of the zerocrossings where the edge strength exceeded some threshold. By using as figure-of-merit the average of the probability of detecting an edge given that it is true and the probability of a true edge given than it is detected, the morphological method scored better by yielding detection probabilities of 0.84 and 0.63 at the noise levels of 20 and 6 dB, respectively, whereas the corresponding probabilities of the LoG method where 0.81 and 0.52.

## 5.2 Peak/Valley Blob Detection

Residuals between openings or closings and the original image offer an intuitively simple and mathematically formal way for peak or valley detection. The general principle for peak detection is to subtract from a signal an opening of it. If the latter is a standard Minkowski opening by a flat compact convex set $B$, then this yields the peaks of the signal whose base cannot contain $B$. 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 Minkowski opening in peak detection is replaced by a general lattice opening, usually of the reconstruction type. This generalization allows a more effective estimation of the image background surroundings around the peak and hence a better detection of the peak. Next we discuss peak detectors based on both the standard Minkowski openings as well as on generalized lattice openings like contrast-based reconstructions which can control the peak height.

### 5.2.1 Top-hat Transformation

Subtracting from a signal $f$ its Minkowski opening by a compact convex set $B$ yields an output consisting of the signal peaks whose supports cannot contain $B$. This is Meyer's *top-hat transformation* [14], implemented by the opening residual

$$\text{peak}(f) \triangleq f - (f \circ B) \tag{50}$$

and henceforth called the **peak** operator. The output peak$(f)$ is always a nonnegative signal, which guarantees that it contains only peaks. Obviously the set $B$ is a very important parameter of the peak operator, because the shape and size of the peak's support obtained by (50) are controlled by the shape and size of $B$. Similarly, to extract the valleys of a signal $f$ we can apply the closing residual

$$\text{valley}(f) \triangleq (f \bullet B) - f \tag{51}$$

henceforth called the **valley** operator.

If $f$ is an intensity image, then the opening (or closing) residual is a very useful operator for detecting *blobs*, defined as a regions with significantly brighter (or darker) intensities relative to the surroundings. Examples are shown in Fig. 10.

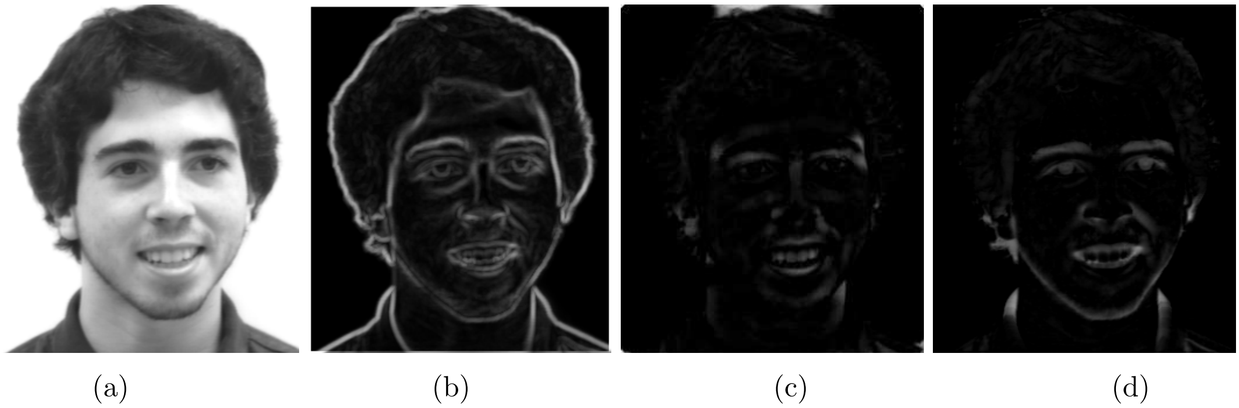

(a) (b) (c) (d)

Figure 10: Facial image feature extraction. (a) Original image $f$. (b) Morphological gradient $f \oplus B - f \ominus B$. (c) Peaks: $f - (f \circ 3B)$. (d) Valleys: $(f \bullet 3B) - f$. ($B$ is 21-pixel octagon.)

If the signal $f(x)$ assumes only the values $0, 1, ..., L$ and we consider its threshold binary signals $f_i(x)$ defined in (44), then since the opening by $f \circ B$ obeys the threshold-sum superposition,

$$\text{peak}(f) = \sum_{i=1}^{L} \text{peak}(f_i)\ . \tag{52}$$

Thus the peak operator obeys threshold-sum superposition. Hence, its output when operating on a graylevel signal $f$ is the sum of its binary outputs when it operates on all the theshold binary versions of $f$. Note that, for each binary signal $f_i$, the binary output peak$(f_i)$ contains only those non-zero parts of $f_i$ inside which no translation of $B$ fits.

The morphological peak and valley operators, in addition to their being simple and efficient, avoid several short-comings of the curvature-based approaches to peak/valley extraction that can be found in earlier computer vision literature. A differential-geometry interpretation of the morphological feature detectors was given by Noble [16], who also developed and analyzed simple operators based on residuals from openings and closings to detect corners and junctions.

### 5.2.2 Dome/Basin Extraction with Reconstruction Opening

Extracting the peaks of a signal via the simple top-hat operator (50) does not constrain the height of the resulting peaks. Specifically, from the threshold-sum superposition of the opening difference in (52) implies that the peak height at each point is the sum of all binary peak signals at this point. In several applications, however, it is desirable to extract from a signal $f$ peaks that have a maximum height $h > 0$. Such peaks are called *domes* and are defined as follows. Subtracting a contrast height constant $h$ from $f(x)$ yields the smaller signal $g(x) = f(x) - h < f(x)$. Enlarging the maximum peak value of $g$ below a peak of $f$ by locally dilating $g$ with a symmetric compact and convex set of an ever-increasing diameter and always restricting these dilations to never produce a signal larger than $f$ under this specific peak produces in the limit a signal which consists of valleys interleaved with flat plateaus. This signal is the *reconstruction opening* of $g$ under $f$, denoted as $\rho^-(g|f)$; namely, $f$ is the reference signal and $g$ is the marker. Subtracting the reconstruction opening from $f$ yields the **domes** of $f$, defined in [28] as the generalized top-hat

$$\text{dome}(f) \triangleq f - \rho^-(f - h|f) \tag{53}$$

For *discrete-domain* signals $f$ the above reconstruction opening can be implemented by iterating the conditional dilation as in (29). This is a simple but computationally expensive algorithm. More efficient algorithms can be found in [28, 21]. The dome operator extracts peaks whose height cannot exceed $h$ but their supports can be arbitrarily wide. In contrast the peak operator (using the opening residual) extracts peaks whose supports cannot exceed a set $B$ but their heights are unconstrained.

Similarly, an operator can be defined that extracts signal valleys whose depth cannot exceed a desired maximum $h$. Such valleys are called **basins** and are defined as the domes of the negated signal: By using the duality between between morphological operations it can be shown that basins of height $h$ can be extracted by subtracting the original image $f(x)$ from its reconstruction closing obtained using as marker the signal $f(x) + h$:

$$\text{basin}(f) \triangleq \text{dome}(-f) = \rho^+(f + h|f) - f \tag{54}$$

Domes and basins have found numerous applications as region-based image features and as markers in image segmentation tasks. Several successful paradigms are discussed in [1, 21, 28].

The following example, adapted from [28], illustrates that domes perform better than the classic top-hat in extracting small isolated peaks that indicate pathology points in biomedical images, e.g. detect micro-aneurisms in eye angiograms without confusing them with the large vessels in the eye image. See Fig. 11.

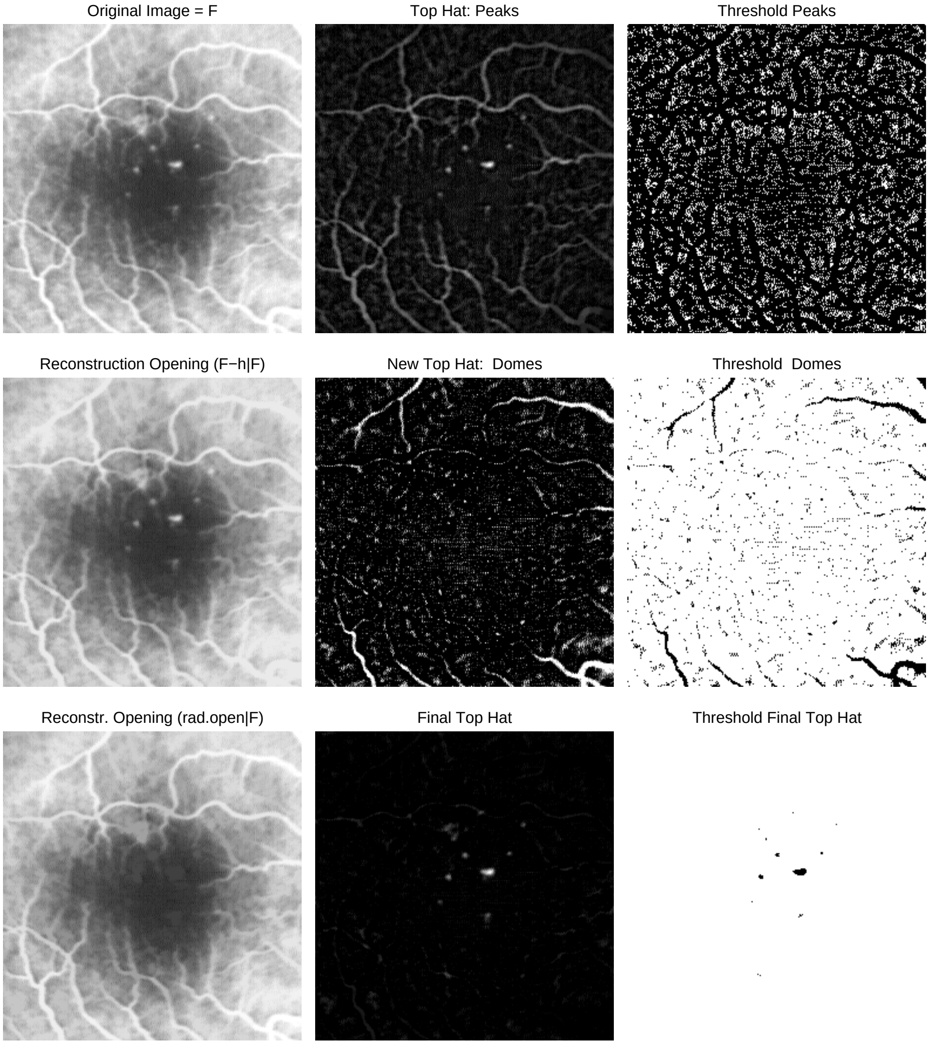


Figure 11: Top row: Original image $F$ of eye angiogram with micro-aneurisms, its top hat $F - F \circ B$ where $B$ is a disk of radius 5, and level set of top hat at height $h/2$. Middle row: Reconstruction opening $\rho^-(F - h|F)$, domes $F - \rho^-(F - h|F)$, level set of domes at height $h/2$. Bottom row: New reconstruction opening of $F$ using the radial opening of Fig.2(b) as marker, new domes and level set detecting micro-aneurisms.

# 6 Optimal Design of Morphological Filters for Enhancement

## 6.1 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 [10]. 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 [9] based on the morphological basis representation theory (outlined in Section 2.3); (b) designing stack filters via threshold decomposition and linear programming [2]; (c) designing morphological networks using either voting logic and rank tracing learning or simulated annealing [30]; (d) designing morphological/rank filters via a gradient-based adaptive optimization [20]. 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 [17], 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. The material in this section is a brief presentation of the work of Pessoa & Maragos [17].

## 6.2 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

$$\begin{aligned} y &\triangleq \lambda\alpha + (1-\lambda)\beta \ , \\ \alpha = \mathcal{R}_r(\underline{x} + \underline{a}) &= \mathcal{R}_r(x_1 + a_1, x_2 + a_2, \cdots, x_n + a_n) \ , \\ \beta = \underline{x} \cdot \underline{b}^T &= x_1 b_1 + x_2 b_2 + \cdots + x_n b_n \ , \end{aligned} \tag{55}$$

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.*, $\mathcal{R}_r(\underline{t}) \triangleq t_{(r)}$ , $r = 1, 2, \cdots, n$. 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 $\lambda$ 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 (55) 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

$$q(v) \triangleq \begin{cases} 1 & , \text{ if } v = 0 \\ 0 & , \text{ otherwise} \end{cases} \tag{56}$$

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

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

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

$$\underline{c}(\underline{t}, r) \triangleq \frac{Q(z\underline{1} - \underline{t})}{Q(z\underline{1} - \underline{t}) \cdot \underline{1}^T} \ , \quad z = \mathcal{R}_r(\underline{t}) \ , \tag{57}$$

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 [17], 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

$$\left. \frac{\partial \mathcal{R}_r(\underline{t})}{\partial \underline{t}} \right|_{\underline{t} = \underline{t}_0} = \underline{c}(\underline{t}_0, r). \tag{58}$$

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:

$$\begin{array}{lll} q_\sigma(v) = q_\sigma(-v) & \text{(symmetry)} & \\ q_\sigma(v) \to q(v) \ \forall \, v & \text{as} & \sigma \to 0, \\ q_\sigma(v) \to 1 \ \forall \, v & \text{as} & \sigma \to \infty. \end{array} \tag{59}$$

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

From the filter definition (55), we see that our design goal is to specify a set of parameters $\underline{a}$, $\underline{b}$, $r$ and $\lambda$ 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 $\rho$ implicitly defined via the following rescaling

$$r \triangleq \left\lfloor n - \frac{n-1}{1+\exp(-\rho)} + 0.5 \right\rfloor , \ \rho \in \mathbb{R} \ , \tag{60}$$

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

$$\underline{w} \triangleq (\underline{a}, \ \rho, \ \underline{b}, \ \lambda) \ , \tag{61}$$

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

## 6.3 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

$$\underline{w}(i+1) = \underline{w}(i) - \mu_0 \nabla J(\underline{w})|_{\underline{w}=\underline{w}(i)} \ , \tag{62}$$

so that the value of the cost function tends to decrease at each step. The positive constant $\mu_0$ is usually called the *step size* and regulates the tradeoff between stability and speed of convergence of the iterative procedure. The iteration (62) 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

$$J(\underline{w}(i)) = \frac{1}{M} \sum_{k=i-M+1}^{i} e^2(k) \ , \tag{63}$$

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

$$e(k) = d(k) - y(k) \tag{64}$$

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

$$\underline{w}(i+1) = \underline{w}(i) + \frac{\mu}{M} \sum_{k=i-M+1}^{i} e(k) \frac{\partial y(k)}{\partial \underline{w}} \bigg|_{\underline{w}=\underline{w}(i)} \ , \ i = 0, 1, 2, \cdots , \tag{65}$$

where $\mu = 2\mu_0$. From (61) and (55)

$$\frac{\partial y}{\partial \underline{w}} = \left( \frac{\partial y}{\partial \underline{a}}, \frac{\partial y}{\partial \rho}, \frac{\partial y}{\partial \underline{b}}, \frac{\partial y}{\partial \lambda} \right) = \left[ \lambda \frac{\partial \alpha}{\partial \underline{a}}, \lambda \frac{\partial \alpha}{\partial \rho}, (1-\lambda)\underline{x}, \alpha - \beta \right] . \tag{66}$$

According to (58) and our design choice, we set

$$\frac{\partial \alpha}{\partial \underline{a}} = \underline{c} = \frac{Q(\alpha \underline{1} - \underline{x} - \underline{a})}{Q(\alpha \underline{1} - \underline{x} - \underline{a}) \cdot \underline{1}^T} , \quad \alpha = \mathcal{R}_r(\underline{x} + \underline{a}). \tag{67}$$

The final unknown is $s = \partial\alpha/\partial\rho$, which will be one more design choice. Notice from (60) and (55) 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 $\alpha$, then variations in the rank $r$ can drastically modify the output $\alpha$; in this case $s$ should assume a maximum value. Thus, a possible simple choice for $s$ is

$$\frac{\partial \alpha}{\partial \rho} = s \triangleq 1 - \frac{1}{n} Q(\alpha \underline{1} - \underline{x} - \underline{a}) \cdot \underline{1}^T , \quad \alpha = \mathcal{R}_r(\underline{x} + \underline{a}) , \tag{68}$$

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 (59)), in which case an appropriate smoothing parameter $\sigma$ 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 (65). We should use small values of $\sigma$ such that $q_\sigma(v)$ is close enough to $q(v)$. A possible systematic way to select the smoothing parameter $\sigma$ could be to set $|q_\sigma(v)| \leq \epsilon$ for $|v| \geq \delta$, so that, for some desired $\epsilon$ and $\delta$, $\sigma = \delta/\sqrt{\ln(1/\epsilon^2)}$.

Theoretical conditions for convergence of the training process (65) can be derived under the following considerations. The goal is to find upper bounds $\mu_w$ to the step size $\mu$, such that (65) 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 (65) to track only the fourth unknown parameter (set) of $\underline{w}^*$ in a noiseless environment. If the training process (65) 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 [17].

## 6.4 Application of Optimal MRL Filters to Enhancement

The proper operation of the training process (65) has been verified in [17] 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.[5] The goal here it to

[5] 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 $q_\sigma(v) = \exp\ -\frac{1}{2}(v/\sigma)^2$ , with $\sigma = 0.001$. The image values are normalized to be in the range $[0, 1]$.

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 (65) 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.

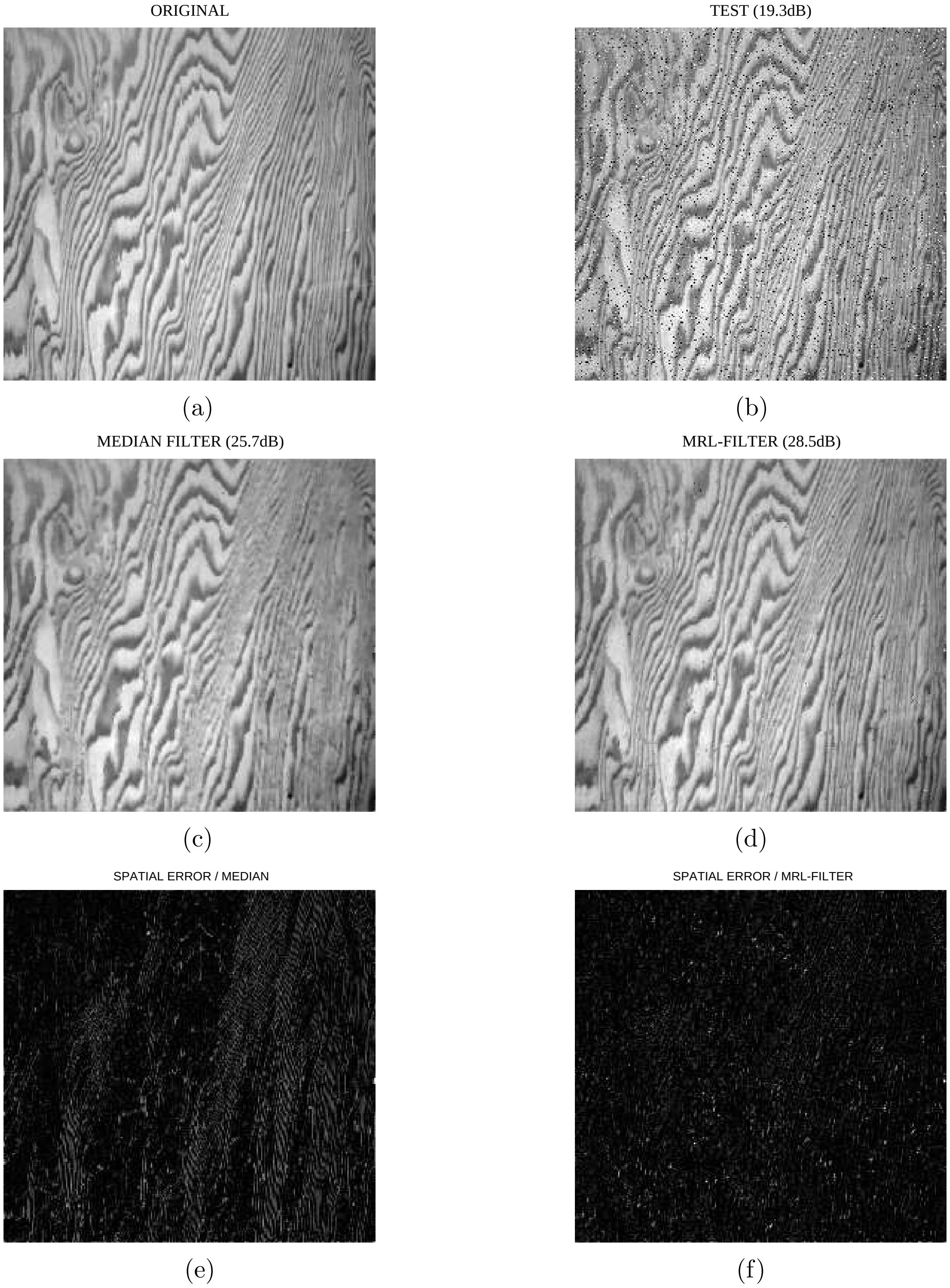
Figure 12 shows the results of using the designed MRL-filter with a test image, and its comparison with a flat median filter of the same window size. The noisy image used for training is not included there because the (noisy) images used for training and testing are simply different realizations of the same perturbation process. Observe that the MRL-filter outperformed the median filter by about 3 dB. Spatial error plots are also included 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 (65). 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 [20]. 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.

# 7 Conclusions

In this chapter we have briefly presented the application of both the standard and some advanced morphological filters to several problems of image enhancement of feature detection. There are



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

Figure 12: (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.

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. Finally, the elementary morphological operators are the building blocks for large classes of nonlinear image processing systems, which include rank and stack filters.

Two important broad research directions in morphological filtering are (1) their optimal design and (2) their scale-space formulation and implementation using geometric partial differential equations (PDEs). In this chapter we have presented some first steps toward (1) from the work in [17]. In Chapter 4.16 of this Handbook we also present their PDE-based formulation and application to multiscale tasks of image analysis and computer vision.

## Acknowledgements

Research support during writing of this chapter was provided in part by the European Network of Excellence 'MUSCLE' and by the Greek research programs PENED-2001 of G.S.R.T. and Pythagoras of the Ministry of Education.

## References

[1] A. Banerji and J. Goutsias, "A Morphological Approach to Automatic Mine Detection Problems", *IEEE Trans. Aerospace and Electron. Systems*, vol.34, p.1085-1096, Oct. 1998.

[2] 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.

[3] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. Pattern Anal. Mach. Intellig.*, vol.PAMI-8, pp.679-698, Nov. 1986.

[4] R. J. Feehs and G. R. Arce, "Multidimensional Morphological Edge Detection", *Proc. SPIE vol. 845: Visual Communications and Image Processing II*, pp.285-292, 1987.

[5] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, Vol.I, Addison-Wesley, 1992.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] D. Marr and E. Hildreth, "Theory of Edge Detection", *Proc. R. Soc. Lond. B*, vol.207, pp.187-217, 1980.

[13] G. Matheron, *Random Sets and Integral Geometry*, Wiley, NY, 1975.

[14] 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.

[15] F. Meyer and P. Maragos, "Nonlinear Scale-Space Representation with Morphological Levelings", *J. Visual Communic. and Image Representation*, vol.11, pp.245-265, 2000.

[16] J. A. Noble, “Morphological Feature Detection”, *Proc. Int’l Conf. Comp. Vision*, Tarpon-Springs, FL, 1988.

[17] 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.

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

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

[20] P. Salembier, “Adaptive Rank Order Based Filters,” *Signal Processing*, vol.27, pp.1-25, 1992.

[21] P. Salembier, “Region-based Filtering of Images and Video Sequences: A Morphological Viewpoint”, in *Nonlinear Image Processing*, edited by S. K. Mitra and G. L. Sicuranza, Acad. Press, 2001.

[22] P. Salembier and J. Serra, “Flat Zones Filtering, Connected Operators, and Filters by Reconstruction”, *IEEE Trans. Image Process.*, vol. 4, pp.1153-1160, Aug. 1995.

[23] 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.

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

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

[26] J. Serra and P. Salembier, “Connected Operators and Pyramids”, in *Image Algebra and Mathematical Morphology*, Proc. SPIE vol.2030, pp.65-76, 1993.

[27] 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.

[28] L. Vincent, “Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms”, *IEEE Trans. Image Proces.*, vol.2, p.176-201, Apr. 1993.

[29] L. J. van Vliet, I. T. Young and G. L. Beckers, “A Nonlinear Laplace Operator as Edge Detector in Noisy Images”, *Comp. Vision, Graphics, and Image Process.*, vol.45, pp.167-195, 1989.

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