Design of Statistically Optimal Stack Filters

N. S. T. Hirata[1], J. Barrera[1], E. R. Dougherty[2]

[1] Instituto de Matemática e Estatística – USP
Rua do Matão, 1010 , 05508-900 São Paulo - SP - Brasil
<nina,jb>@ime.usp.br
[2] Department of Electrical Engineering – Texas A & M University
214 Wisenbaker Engineering Research Center
College Station – TX – 77843-3407, US
edward@ee.tamu.edu

Abstract. Any gray-scale image can be represented as a "stack" of a decreasing sequence of binary images, obtained by thresholding the gray-scale image at each level. Stack filters are a special class of gray-scale image operators whose filtered images can be represented as the stack of binary images resulting of applying an increasing binary operator for each image of the stack. A classical example of stack filter is the median filter: the median of a gray-scale image is the same as the sum of the binary median computed for each of the binary images in the stack. Thus, the design of stack filters can be reduced to the design of the binary operators that characterize them. This paper reviews stack filters in the context of mathematical morphology on discrete images, and shows that the problem of designing optimal mean-absolute error (MAE) stack filters is an optimization problem equivalent to the design of optimal MAE binary increasing operators. A new combinatorial algorithm for designing binary increasing operators is applied on the design of stack filters for impulse and speckle noise reduction.

1 Introduction

Systematic and automatic design of image operators is an important requirement for many application fields that deal with digital images. In recent years, some efforts have contributed to the development of automatic methods for designing translation invariant morphological operators that depend on an observation window W (i.e., *W-operators*) [1, 2]. These methods consist usually of machine-learning based approaches, where training images are used for the estimation of a W-operator with best performance according to some error criterion [1]. However, the optimization over the entire class of W-operators, for large windows, is computationally hard due to the size of this space. For a window of size n, there are exactly $2^{(2^n)}$ possible binary W-operators. Another difficulty is related to the precision of the estimated operators [3]. For large windows, the amount of data needed for good estimations is so large that often it is impossible to obtain an operator with an acceptable precision.

One of the possible approaches to deal with these difficulties is the reduction of the search space. This can be achieved, for instance, by imposing that the operators must satisfy some algebraic property. The class of increasing W-operators have been largely studied due to its relatively simple representation, implying also implementation simplicity. The authors have proposed recently a new algorithm for the design of binary increasing W-operators, based on switching of the optimal operator in such a way to generate an increasing operator [4].

The design of operators for gray-scale images is much harder, because the number of possible gray-scale W-operators is $k^{(k^n)}$, where k is the number of image gray-scales. There exists, however, a special class of gray-scale W-operators that can be represented in terms of their corresponding binary W-operators, such as the median filter. The median [5] of a gray-scale image is equal to the sum of the binary median applied to each of the binary images that are obtained by thresholding the gray-scale image at each level. The class of gray-scale W-operators having the above characteristic form the class of *stack filters* [6]. In fact, stack filters are characterized by positive Boolean functions (or equivalently, increasing binary W-operators). Consequently, an algorithm for the design of increasing binary operators may be useful for designing stack filters.

This paper reviews the class of stack filters in the context of mathematical morphology and discrete images (i.e., with discrete domain and range), and shows how the problem of designing optimal MAE stack filters can be viewed as a problem of designing optimal MAE increasing binary W-operators. In particular, we show the use of the switching algorithm for this pur-

pose.

Following this introduction, section 2 presents the class of stack filters and shows their correspondence to the class of positive Boolean functions. Section 3 reviews the representation of stack filters in terms of erosions. Section 4 discusses the notion of optimal stack filters relative to the MAE criterion, and defines an optimization problem in terms of the probabilities of the random sets associated to each of the threshold levels. Section 5 presents a review of the switching algorithm that may be used to solve the optimization problem stated on section 4. Section 6 discusses estimation, from training images, of the costs needed for the application of the switching algorithm for the design of stack filters. Section 7 presents examples of stack filter application for impulse and speckle noise filtering. Complete proofs for propositions stated without proof can be found in [7].

2 Stack Filters

Let $E = Z^2$ and $K = \{0, 1, \ldots, k\}$, where k is a positive integer. A function f from E to K is a *gray-scale image*. The set of all gray-scale images is denoted by K^E. For any $f, g \in K^E$, $f \leq g \Leftrightarrow f(x) \leq g(x), \forall x \in E$.

Let $\mathcal{P}(E)$ denote the power set of E. The *image threshold* at level $t \in K$ is a mapping T_t from K^E to $\mathcal{P}(E)$, given by, for any $f \in K^E$ and $t \in K$,

$$T_t[f] = \{x \in E : f(x) \geq t\}. \tag{1}$$

The set $T_t[f]$ is called the *cross-section* of f at level t. The *value threshold* of an element in K at level $t \in K$ is a mapping from K to $\{0, 1\}$, given by, for any $v \in K$ and $t \in K$,

$$T_t(v) = 1 \Leftrightarrow v \geq t. \tag{2}$$

Note that $T_t[f]$ denotes a subset of E, while $T_t(v)$ denotes a binary value.

Proposition 2.1 *Let $t_1, t_2 \in K$ and $f \in K^E$. If $t_1 \leq t_2$, then $T_{t_2}[f] \subseteq T_{t_1}[f]$.*

Proposition 2.2 *Given $f, g \in K^E$, $f \leq g \Leftrightarrow T_t[f] \subseteq T_t[g], \forall t \in K$.*

A binary image is an element of $\{0, 1\}^E$. Each binary image defines a unique subset of E and vice-versa. Given a binary image $f \in \{0, 1\}^E$, the corresponding subset is $T_1[f]$. Given a subset $S \subseteq E$, the corresponding binary image is the function 1_S given by, for any $x \in E$,

$$1_S(x) = 1 \Leftrightarrow x \in S. \tag{3}$$

One can verify that, for any $f \in \{0, 1\}^E$ and $X \subseteq E$, $1_{T_1[f]} = f$ and $T_1[1_X] = X$. Moreover, $X \subseteq Y \Leftrightarrow 1_X \leq 1_Y$.

If $S_1, S_2, \ldots, S_m$ is a sequence of m subsets of E, we say that it obeys the *stacking property* if

$$S_m \subseteq \ldots \subseteq S_2 \subseteq S_1. \tag{4}$$

The cross sections $T_t[f]$, $t \in K$, form a sequence of subsets characterizing binary images. Any gray-scale image $f \in K^E$ can be represented as the sum of a sequence of binary images, i.e., for any $x \in E$,

$$f(x) = \sum_{t=1}^{k} 1_{T_t[f]}(x). \tag{5}$$

Moreover, since the cross-sections of f obey the stacking property (Proposition 2.1), it is also true that

$$f(x) = max\{t \in K : x \in T_t[f]\}. \tag{6}$$

Formulas 1, 3 to 6, and Proposition 2.1 and 2.2 are also valid when E is changed by a finite subset $W \subseteq E$.

From now on, we consider a finite subset $W \subseteq E$, $W = \{w_1, w_2, \ldots, w_n\}$, called *window*. The set of functions from W to K is denoted K^W. The translation of $X \subseteq E$ by a vector $z \in E$ is denoted X_z and defined by $X_z = \{x + z : x \in X\}$. The *restriction* of $f \in K^E$ to W_x defines a function $f|W_x$ in K^W given by $(f|W_x)(w) = f(x + w)$, for any $w \in W$. The *translation* of f by a vector $z \in E$ is denoted f_z and defined by, $\forall x \in E$, $f_z(x) = f(x - z)$.

An image operator $\Psi : K^E \to K^E$ is :

- *translation-invariant* (t.i.) if and only if (iff), for any $f \in K^E$ and $z \in E$,

$$\Psi(f_z) = [\Psi(f)]_z \tag{7}$$

- *locally defined* (l.d.) within W iff, for any $f \in K^E$ and $x \in E$,

$$\Psi(f)(x) = \Psi(g)(x) \tag{8}$$

  for any $g \in K^E$ such that $f|W_x = g|W_x$.

- *increasing* iff, for any $f, g \in K^E$, if $f \leq g$, then $\Psi(f) \leq \Psi(g)$.

An image operator $\Psi : K^E \to K^E$ is a *W-operator* iff it is both t.i. and l.d. A W-operator $\Psi : K^E \to K^E$ can be uniquely characterized by a function $\psi : K^W \to K$ [8], i.e., for any $x \in E$ and $f \in K^E$,

$$\Psi(f)(x) = \psi(f|W_x). \tag{9}$$

Given an image operator $\Psi : K^E \to K^E$, let $\tilde{\Psi} : \mathcal{P}(E) \to \mathcal{P}(E)$ be the binary image operator (also called *set operator*) defined by, for any $X \in \mathcal{P}(E)$,

$$\tilde{\Psi}(X) = T_1[\Psi(1_X)]. \tag{10}$$

Similarly, given $\psi : K^W \to K$, let $\tilde{\psi} : \mathcal{P}(W) \to \{0,1\}$ be the Boolean function defined by, for any $X \in \mathcal{P}(W)$,

$$\tilde{\psi}(X) = min\{1, \psi(1_X|W)\}\,. \tag{11}$$

Definition 2.1 *An image operator $\Psi : K^E \to K^E$ is a* stack filter *iff*
1. Ψ is a W-operator
2. Ψ commutes with threshold, i.e., for any $t \in K$ and $f \in K^W$,

$$T_t(\psi(f)) = \tilde{\psi}(T_t[f])\,. \tag{12}$$

Proposition 2.3 *If $\Psi : K^E \to K^E$ is a stack filter, then ψ and Ψ are increasing* (see [9] for a proof.)

Proposition 2.4 *The set of stack filters and the set of positive (i.e., increasing) Boolean functions are isomorphic.*

Proof : First, we will define two mappings, α from the set of positive Boolean functions to the set of stack filters, and β from the set of stack filters to the set of positive Boolean functions (see also Fig. 1). Second, we will show that they define a lattice isomorphism.

For any positive Boolean function $\tilde{\phi}$, $f \in K^E$ and $x \in E$, let

$$\Big(\alpha(\tilde{\phi})(f)\Big)(x) = a(\tilde{\phi})(f|W_x) \tag{13}$$

where, for any $g \in K^W$,

$$a(\tilde{\phi})(g) = max\{t \in K : \tilde{\phi}(T_t[g]) = 1\}\,. \tag{14}$$

For any stack filter Ψ (with respective characteristic function ψ) and $X \subseteq W$, let

$$\beta(\Psi)(X) = \tilde{\psi}(X)\,. \tag{15}$$

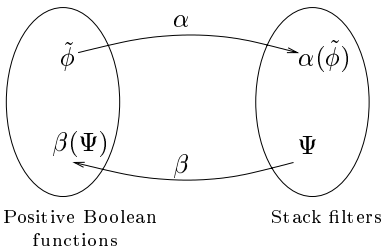

Figure 1: Isomorphism between positive Boolean functions and stack filters.

a) $\alpha(\tilde{\phi})$ is a W-operator by construction. We show that it commutes with threshold too. First, we show that the Boolean function $\tilde{a}(\tilde{\phi})$ corresponding to $a(\tilde{\phi})$ is $\tilde{\phi}$ itself. For any $X \in \mathcal{P}(W)$,

$$\begin{aligned}
&\tilde{a}(\tilde{\phi})(X) = \\
&\quad \overset{(1)}{=} \; min\{1, a(\tilde{\phi})(1_X|W)\} \\
&\quad \overset{(2)}{=} \; min\{1, max\{t \in K : \tilde{\phi}(T_t[1_X|W]) = 1\} \\
&\quad \overset{(3)}{=} \; \begin{cases} 1, & \text{if } \tilde{\phi}(\emptyset) = 1, \\ 0, & \text{if } \tilde{\phi}(W) = 0, \\ \tilde{\phi}(T_1[1_X|W]), & \text{otherwise.} \end{cases} \\
&\quad \overset{(4)}{=} \; \begin{cases} 1, & \text{if } \tilde{\phi}(\emptyset) = 1, \\ 0, & \text{if } \tilde{\phi}(W) = 0, \\ \tilde{\phi}(X), & \text{otherwise.} \end{cases} \\
&\quad \overset{(5)}{=} \; \tilde{\phi}(X)
\end{aligned}$$

Equality (1) follows from Eq. 11, (2) from Eq. 14, (3) from the fact that $T_t[1_X|W] = \emptyset$ for any $t \geq 2$, and (4) from the fact that $T_1[1_X|W] = X$. Hence, for any $g \in K^W$ and $t \in K$,

$$\begin{aligned}
&T_t(a(\tilde{\phi})(g)) = 1 \Leftrightarrow \\
&\quad \overset{(1)}{\Leftrightarrow} \; a(\tilde{\phi})(g) \geq t \\
&\quad \overset{(2)}{\Leftrightarrow} \; max\{i \in K : \tilde{\phi}(T_i[g]) = 1\} \geq t \\
&\quad \overset{(3)}{\Leftrightarrow} \; max\{i \in K : \tilde{a}(\tilde{\phi})(T_i[g]) = 1\} \geq t \\
&\quad \overset{(4)}{\Leftrightarrow} \; \tilde{a}(\tilde{\phi})(T_t[g]) = 1
\end{aligned}$$

Equivalence (1) follows from Eq. 2, (2) from Eq. 14, (3) from the fact that $\tilde{a}(\tilde{\phi}) = \tilde{\phi}$, and (4) from Eq. 14.
b) We show that $\beta(\Psi)$ is a positive Boolean function. Let $X, Y \subseteq W$. If $X \subseteq Y$, then

$$\begin{aligned}
\beta(\Psi)(X) &\overset{(1)}{=} min\{1, \psi(1_X|W)\} \\
&\overset{(2)}{\leq} min\{1, \psi(1_Y|W)\} \\
&\overset{(3)}{=} \beta(\Psi)(Y)
\end{aligned}$$

Equalities (1) and (3) follow from Eq. 15 and 11, and (2) from the fact that ψ is increasing (Proposition 2.3) and $X \subseteq Y \Leftrightarrow 1_X|W \leq 1_Y|W$.
c) The mappings α and β define a bijection. On one hand, $\alpha(\beta(\Psi)) = \Psi$ for any stack filter Ψ, as shown below.

$$\begin{aligned}
&\Big(\alpha(\beta(\Psi))(f)\Big)(x) = \\
&\quad \overset{(1)}{=} \; max\{t \in K : \beta(\Psi)(T_t[f|W_x]) = 1\} \\
&\quad \overset{(2)}{=} \; max\{t \in K : min\{1, \psi(1_{T_t[f|W_x]})\} = 1\} \\
&\quad \overset{(3)}{=} \; max\{t \in K : \psi(1_{T_t[f|W_x]}) \geq 1\} \\
&\quad \overset{(4)}{=} \; max\{t \in K : T_1(\psi(1_{T_t[f|W_x]})) = 1\} \\
&\quad \overset{(5)}{=} \; max\{t \in k : \tilde{\psi}(T_1[1_{T_t[f|W_x]}]) = 1\}
\end{aligned}$$

$$\begin{aligned}
&\stackrel{(6)}{=} max\{t \in K : \tilde{\psi}(T_t[f|W_x]) = 1\} \\
&\stackrel{(7)}{=} max\{t \in K : T_t(\psi(f|W_x)) = 1\} \\
&\stackrel{(8)}{=} \psi(f|W_x) \\
&= \Psi(f)(x)
\end{aligned}$$

Equality (1) follows from Eq. 13 and 14, (2) from Eq. 15, (3) from property of min, (4) because $T_1(v) = 1 \Leftrightarrow v \geq 1$, (5) and (7) because ψ commutes with threshold, (6) because $T_1[1_X] = X$; and (8) from Eq. 6 for values.

On the other hand, $\beta(\alpha(\tilde{\phi})) = \tilde{\phi}$, for any positive Boolean function $\tilde{\phi}$, as shown below.

$$\begin{aligned}
&\beta(\alpha(\tilde{\phi}))(X) = \\
&\stackrel{(1)}{=} min\{1, a(\tilde{\phi})(1_X|W)\} \\
&\stackrel{(2)}{=} min\{1, max\{t \in K : \tilde{\phi}(T_t[1_X|W]) = 1\}\} \\
&\stackrel{(3)}{=} \begin{cases} 1, & \text{if } \tilde{\phi}(\emptyset) = 1, \\ 0, & \text{if } \tilde{\phi}(W) = 0, \\ \tilde{\phi}(T_1[1_X|W])), & \text{otherwise.} \end{cases} \\
&= \tilde{\phi}(X)
\end{aligned}$$

Equality (1) follows from Eq. 15 and the fact that $a(\tilde{\phi})$ is the function that characterizes $\alpha(\tilde{\phi})$, (2) from Eq. 13 and 14, and (3) from the fact that $T_1[1_X|W] = X$.
d) The proof that α and β preserve the partial order relation follows directly from their definition. □

Proposition 2.4 means that each stack filter is uniquely characterized by a positive Boolean function and, conversely, each positive Boolean function uniquely characterizes a stack filter. The mappings α and β show, respectively, how to build the stack filter corresponding to a given positive Boolean function and how to build a positive Boolean function corresponding to a given stack filter.

3 Representation of stack filters

In this section, we show that stack filters can be represented as a maximum of erosions by flat structuring elements. Moreover, given the minimal sum of products decomposition of the Boolean function that characterizes the stack filter, we show how to determine these structuring elements.

Definition 3.1 *Let $B \subseteq E$, B finite. The mapping $E_B : K^E \rightarrow K^E$ given by, for any $f \in K^E$ and $x \in E$,*

$$E_B(f)(x) = \varepsilon_B(f|W_x) \tag{16}$$

where, for any $g \in K^W$,

$$\varepsilon_B(g) = min\{g(y) : y \in B\} \tag{17}$$

is called the erosion of f by the set B.

If $B \subseteq W$, then E_B is a W-operator. The set operator corresponding to E_B is the mapping $\tilde{E}_B : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$, given by

$$x \in \tilde{E}_B(X) \Leftrightarrow E_B(1_X)(x) = 1\,. \tag{18}$$

The Boolean function $\tilde{\varepsilon}_B$ corresponding to ε_B is given by, for any $X \subseteq W$,

$$\tilde{\varepsilon}_B(X) = 1 \Leftrightarrow \varepsilon_B(1_X|W) = 1 \Leftrightarrow B \subseteq X \tag{19}$$

Proposition 3.1 *The erosion operator defined by Eq. 16 commutes with threshold.*

Definition 3.2 *The* kernel *of a binary W-operator $\tilde{\Psi}$, with characteristic function $\tilde{\psi}$, is given by*

$$\mathcal{K}(\tilde{\Psi}) = \{X \in \mathcal{P}(W) : \tilde{\psi}(X) = 1\} \tag{20}$$

Definition 3.3 *The* basis *of an increasing binary W-operator $\tilde{\Psi}$ is denoted $\mathcal{B}(\tilde{\Psi})$ and defined as the set of minimal elements of $\mathcal{K}(\tilde{\Psi})$ (i.e., if $X \in \mathcal{B}(\tilde{\Psi})$, then $X' \subseteq X$ implies $X' = X$, for all $X' \in \mathcal{K}(\tilde{\Psi})$).*

A mapping $\tilde{\psi} : \mathcal{P}(W) \rightarrow \{0,1\}$ is a Boolean function on n variables $x_1, x_2, \ldots, x_n$. The value of $\tilde{\psi}$ for a given input $X \in \mathcal{P}(W)$ is computed by making $x_i = 1 \Leftrightarrow w_i \in X$, for all $i \in \{1, 2, \ldots, n\}$, and evaluating the logical expression representing $\tilde{\psi}$.

It is well known that positive Boolean functions can be represented as a logical sum of products, where no complemented variable appears. More specifically,

$$\tilde{\psi} = \sum_{j \in J} m_j \tag{21}$$

where J is a set of indices, $\sum$ represents a logical sum, and the terms $m_j : \mathcal{P}(W) \rightarrow \{0,1\}$ are prime implicants of $\tilde{\psi}$. Each of the prime implicants is a product term with no complemented variable

$$m_j = \prod_{i \in I_j} x_i, \quad I_j \subseteq \{1, 2, \ldots, n\} \tag{22}$$

Hence, $m_j(X) = 1 \Leftrightarrow (w_i \in X, \forall i \in I_j)$. If we define $B_j = \{w_i : i \in I_j\}$, then $m_j(X) = 1 \Leftrightarrow B_j \subseteq X$. Hence, $m_j(X) = 1 \Leftrightarrow \tilde{\varepsilon}_{B_j}(X) = 1$.

Proposition 3.2 *If $\tilde{\psi} : \mathcal{P}(W) \rightarrow \{0,1\}$ is a positive Boolean function, then, for any $X \in \mathcal{P}(W)$,*

$$\tilde{\psi}(X) = max\{\tilde{\varepsilon}_B(X) : B \in \mathcal{B}(\tilde{\Psi})\}\,. \tag{23}$$

Proof : From the discussion above, any positive Boolean function can be expressed in terms of a set of binary erosions. We just need to show that the set

$\{B_j : j \in J\}$ (which corresponds to the set of all prime implicants of $\tilde{\psi}$) and $\mathcal{B}(\tilde{\Psi})$ are the same.

Let $B \in \mathcal{B}(\tilde{\Psi})$. Since $\mathcal{B}(\tilde{\Psi}) \subseteq \mathcal{K}(\tilde{\Psi})$, we must have $\tilde{\psi}(B) = 1$. Hence, there exists $j \in J$ such that $m_j(B) = 1$, what means that $B_j \subseteq B$. We can not have $B_j \subset B$ because B is a minimal element in $\mathcal{K}(\tilde{\Psi})$. Hence, $B = B_j$, $j \in J$, i.e., $B \in \{B_j : j \in J\}$. On the other hand, let $B_i \in \{B_j : j \in J\}$. Since $m_i(B_i) = 1$, it follows that $\tilde{\psi}(B_i) = 1$, and therefore $B_i \in \mathcal{K}(\tilde{\Psi})$. Then, there exists $B \in \mathcal{B}(\tilde{\Psi})$ such that $B \subseteq B_i$, but we can not have $B \subset B_i$ because that would imply that m_i is not a prime implicant of $\tilde{\psi}$. Hence, $B_i = B$, $B \in \mathcal{B}(\tilde{\Psi})$. □

Proposition 3.3 *If $\Psi : K^E \to K^E$ is a stack filter, then, for any $x \in E$ and $f \in K^E$,*

$$\Psi(f)(x) \quad = \quad max\{\varepsilon_B(f|W_x) : B \in \mathcal{B}(\tilde{\Psi})\} \quad (24)$$

Proof : Since $\Psi(f)(x) = \psi(f|W_x)$, $\forall f \in K^E$ and $x \in E$, we just need to show that

$$\psi(f|W_x) \quad = \quad max\{\varepsilon_B(f|W_x) : B \in \mathcal{B}(\tilde{\Psi})\}$$

In fact, $\forall g \in K^W$,

$$\begin{aligned}
\psi(g) = & \\
& \stackrel{(1)}{=} max\{y \in K : T_y(\psi(g)) = 1\} \\
& \stackrel{(2)}{=} max\{y \in K : \tilde{\psi}(T_y[g]) = 1\} \\
& \stackrel{(3)}{=} max\{y \in K : max\{\tilde{\varepsilon}_B(T_y[g]) : B \in \mathcal{B}(\tilde{\Psi})\} = 1\} \\
& \stackrel{(4)}{=} max\{y \in K : max\{T_y(\varepsilon_B(g)) : B \in \mathcal{B}(\tilde{\Psi})\} = 1\} \\
& = max\{max\{y \in K : T_y(\varepsilon_B(g)) = 1\} : B \in \mathcal{B}(\tilde{\Psi})\} \\
& \stackrel{(5)}{=} max\{max\{y \in K : \varepsilon_B(g) \geq y\} : B \in \mathcal{B}(\tilde{\Psi})\} \\
& = max\{\varepsilon_B(g) : B \in \mathcal{B}(\tilde{\Psi})\}
\end{aligned}$$

Equality (1) follows from Eq. 6, (2) because ψ commutes with threshold, (3) from Proposition 3.2, (4) from Proposition 3.1, and (5) from Eq. 2. □

4 Optimal stack filters

The optimal stack filter is one that, among all stack filters, minimizes some error criterion. The mean-absolute-error (MAE) criterion has been largely used since it has the nice property that the MAE of a stack filter Ψ is the sum of the MAE of its corresponding Boolean function over all the cross sections [10].

Let f be an image to be filtered and f_0 be its respective ideal image. The MAE of a stack filter Ψ is given by

$$MAE\langle\Psi\rangle = E\Big[|f_0(z) - \psi(f|W_z)|\Big]. \quad (25)$$

The MAE of the corresponding positive Boolean function $\tilde{\psi}$ at level i is given by

$$MAE_i\langle\tilde{\psi}\rangle = E\Big[|T_i(f_0(z)) - \tilde{\psi}(T_i[f|W_z])|\Big]. \quad (26)$$

Proposition 4.1 *Let Ψ be a stack filter and $\tilde{\psi}$ the corresponding positive Boolean function. The following equality is true (see [10] for a proof.)*

$$MAE\langle\Psi\rangle = \sum_{t=1}^{k} MAE_t\langle\tilde{\psi}\rangle \quad (27)$$

If f and f_0 are jointly stationary, then the value $T_t(f_0(z))$ can be regarded as a random realization of a binary random variable $\tilde{Y}$ and the set $T_t[f|W_z]$ as a random realization of a random set $\tilde{X}$. Let $P_i(\mathrm{x}) = P_i(\tilde{X} = \mathrm{x})$ denote the probability of $\tilde{X} = \mathrm{x}$ at level i, and $P_i(y|\mathrm{x}) = P_i(\tilde{Y} = y|\tilde{X} = \mathrm{x})$ the conditional probability of $\tilde{Y} = y$ given that $\tilde{X} = \mathrm{x}$ at level i, where $\mathrm{x} \in \mathcal{P}(W)$ and $y \in \{0, 1\}$. Hence, Eq. 26 can be rewritten as

$$MAE_i\langle\tilde{\psi}\rangle = \sum_{\mathrm{x}} P_i(\mathrm{x}) \sum_{y} |y - \tilde{\psi}(\mathrm{x})| P_i(y|\mathrm{x}). \quad (28)$$

By expanding Eq. 27, we obtain

$$\begin{aligned}
MAE\langle\Psi\rangle = & \quad (29) \\
& = \sum_{i=1}^{k} \sum_{\mathrm{x}} P_i(\mathrm{x}) \sum_{y} |y - \tilde{\psi}(\mathrm{x})| P_i(y|\mathrm{x}) \\
& = \sum_{\mathrm{x}} \sum_{i=1}^{k} P_i(\mathrm{x}) \sum_{y} |y - \tilde{\psi}(\mathrm{x})| P_i(y|\mathrm{x}).
\end{aligned}$$

Hence, minimization of Eq. 25 is equivalent to minimization of Eq. 29. However, no simple solution to this problem is known. The difficult lies on the fact that $\tilde{\psi}$ must be increasing, meaning that the terms $M_{\mathrm{x}}(\tilde{\psi}) = \sum_{i=1}^{k} P_i(\mathrm{x}) \sum_y |y - \tilde{\psi}(\mathrm{x})| P_i(y|\mathrm{x})$ can not be minimized independently; if $\tilde{\psi}(\mathrm{x})$ is set to 1, then $\tilde{\psi}(\mathrm{y})$ for any $\mathrm{y} > \mathrm{x}$ must be also set to 1. Conversely, if $\tilde{\psi}(\mathrm{x})$ is set to 0, then $\tilde{\psi}(\mathrm{y})$ for any $\mathrm{y} < \mathrm{x}$ must be also set to 0.

Let $M(\tilde{\psi}) = \sum_{i=1}^{k} MAE_i\langle\tilde{\psi}\rangle$. If we consider the problem of determining a (not necessarily positive) Boolean function that minimizes M, then M is minimized when M_{x} is minimized for each x. Since $y \in \{0, 1\}$,

$$\begin{aligned}
M_{\mathrm{x}}(\tilde{\psi}) \quad = \quad & \sum_{i=1}^{k} P_i(\mathrm{x})\, \tilde{\psi}(\mathrm{x})\, P_i(0|\mathrm{x}) \\
& + \sum_{i=1}^{k} P_i(\mathrm{x})\, (1 - \tilde{\psi}(\mathrm{x}))\, P_i(1|\mathrm{x}).
\end{aligned}$$

If we set $\tilde{\psi}(\mathrm{x}) = 0$ then the error incurred by this choice is $c_0(\mathrm{x}) = \sum_{i=1}^{k} P_i(\mathrm{x})\, P_i(1|\mathrm{x})$. On the other hand, if we set $\tilde{\psi}(\mathrm{x}) = 1$ then the error incurred by this choice is $c_1(\mathrm{x}) = \sum_{i=1}^{k} P_i(\mathrm{x})\, P_i(0|\mathrm{x})$. Hence, $M(\tilde{\psi})$ is minimized by the following Boolean function :

$$\tilde{\psi}_{opt}(\mathrm{x}) = \begin{cases} 1, & \text{if } c_0(\mathrm{x}) > c_1(\mathrm{x}), \\ 0, & \text{if } c_0(\mathrm{x}) \leq c_1(\mathrm{x}). \end{cases} \tag{30}$$

Note that, the Boolean function given by Eq. 30 may not be positive.

Given a Boolean function $\tilde{\psi}$, the difference

$$\Delta_M(\tilde{\psi}, \tilde{\psi}_{opt}) = M(\tilde{\psi}) - M(\tilde{\psi}_{opt}) \tag{31}$$

is called the *error increase of* $\tilde{\psi}$ *in relation to* $\tilde{\psi}_{opt}$. The amount that x contributes to $\Delta_M(\tilde{\psi}, \tilde{\psi}_{opt})$ is given by

$$\begin{aligned} \delta(\mathrm{x}) &= M_{\mathrm{x}}(\tilde{\psi}) - M_{\mathrm{x}}(\tilde{\psi}_{opt}) \\ &= c_1(\mathrm{x})[\tilde{\psi}(\mathrm{x}) - \tilde{\psi}_{opt}(\mathrm{x})] + c_0(\mathrm{x})[\tilde{\psi}_{opt}(\mathrm{x}) - \tilde{\psi}(\mathrm{x})] \\ &= \begin{cases} c_1(\mathrm{x}) - c_0(\mathrm{x}), \\ \qquad \text{if } \tilde{\psi}_{opt}(\mathrm{x}) = 0 \text{ and } \tilde{\psi}(\mathrm{x}) = 1, \\ c_0(\mathrm{x}) - c_1(\mathrm{x}), \\ \qquad \text{if } \tilde{\psi}_{opt}(\mathrm{x}) = 1 \text{ and } \tilde{\psi}(\mathrm{x}) = 0, \\ 0, \\ \qquad \text{if } \tilde{\psi}_{opt}(\mathrm{x}) = \tilde{\psi}(\mathrm{x}). \end{cases} \end{aligned}$$

Hence, finding a positive Boolean function that minimizes Eq. 25 is equivalent to finding a positive Boolean function $\tilde{\psi}$ that minimizes $\Delta_M(\tilde{\psi}, \tilde{\psi}_{opt}) = \sum_{\mathrm{x}:\tilde{\psi}_{opt}(\mathrm{x}) \neq \tilde{\psi}(\mathrm{x})} c(\mathrm{x})$, where $\tilde{\psi}_{opt}$ is given by Eq. 30 and the costs $c(\mathrm{x})$ are given by, for any $\mathrm{x} \in \mathcal{P}(W)$,

$$c(\mathrm{x}) = \begin{cases} c_1(\mathrm{x}) - c_0(\mathrm{x}), & \text{if } \tilde{\psi}_{opt}(\mathrm{x}) = 0, \\ c_0(\mathrm{x}) - c_1(\mathrm{x}), & \text{if } \tilde{\psi}_{opt}(\mathrm{x}) = 1. \end{cases} \tag{32}$$

Note that, if $\tilde{\psi}$ is increasing, then $M(\tilde{\psi}) = MAE\langle\Psi\rangle$, where Ψ is the stack filter corresponding to $\tilde{\psi}$.

5 Optimal increasing set operators

Let X be a random set whose realizations are elements of $\mathcal{P}(W)$ and Y be a random variable taking values in $\{0, 1\}$, and suppose (X, Y) is a jointly stationary process. An optimal set operator ψ_{opt}, relative to some error criterion M, is a mapping $\psi : \mathcal{P}(W) \to \{0, 1\}$ that minimizes the expected value $E[d_M(\psi(X), Y)] = M(\psi)$, where $d_M(\psi(X), Y)$ is the function that measures the difference between $\psi(X)$ and Y with relation to M. For instance, $d_{MAE}(\psi(X), Y) = |\psi(X) - Y|$.

Since ψ_{opt} has the smallest error M among all set W-operators, any W-operator ψ has a nonnegative error increase with relation to the error of ψ_{opt}, which is given by

$$\Delta_M(\psi, \psi_{opt}) = M(\psi) - M(\psi_{opt}) \tag{33}$$

An optimal increasing set W-operator is one that, among all increasing set W-operators, possesses the smallest error M. Hence, an optimal increasing W-operator is one that, among all increasing operators, possesses the smallest error increase with relation to ψ_{opt}.

Let $c(\mathrm{x})$ be the *switching cost* (i.e., the amount x contributes to the increase of Δ_M) if the value of $\psi_{opt}(\mathrm{x})$ is switched (from 1 to 0 or from 0 to 1). For the particular case where the error criterion is the MAE, the switching cost is given by $c(\mathrm{x}) = |2\, p_{\mathrm{x}} - 1|\, P(\mathrm{x})$, where $p_{\mathrm{x}} = P(Y = 1|X = \mathrm{x})$ and $P(\mathrm{x}) = P(X = \mathrm{x})$ (see [4]). For the stack filters, costs are given by Eq. 32.

In the remaining of this section, an algorithm, to be called *switching algorithm*, that, given an optimal operator ψ_{opt} and switching costs $c(\mathrm{x})$, computes an increasing operator ψ that minimizes $\Delta_M(\psi, \psi_{opt})$, will be presented.

Let $\psi\langle 0\rangle = \{\mathrm{x} \in \mathcal{P}(W) : \psi(\mathrm{x}) = 0\}$ be the 0-set of ψ, and $\psi\langle 1\rangle = \{\mathrm{x} \in \mathcal{P}(W) : \psi(\mathrm{x}) = 1\}$ be the 1-set of ψ. For any W-operator ψ, its *inversion set*, to be denoted by Q_ψ, is the set consisting of all 1-set elements having at least one 0-set element above them, and all 0-set elements having at least one 1-set element beneath them (with relation to the set inclusion $\subseteq$). ψ is increasing if and only if $Q_\psi = \emptyset$. Hence, an increasing operator can be obtained from a W-operator ψ by switching the value of ψ for some elements of Q_ψ, until the inversion set of the resulting operator becomes empty.

Given $A \subseteq \mathcal{P}(W)$, let $A_\psi\langle 1\rangle = A \cap \psi\langle 1\rangle$ and $A_\psi\langle 0\rangle = A \cap \psi\langle 0\rangle$. If $A \subseteq Q_\psi$ denote the set of elements where ψ will be switched, then the resulting operator (from the switchings) is denoted ψ_A and given by, for any $\mathrm{x} \in \mathcal{P}(W)$,

$$\psi_A(\mathrm{x}) = \begin{cases} 1, & \text{if } \mathrm{x} \in (\psi\langle 1\rangle \setminus A_\psi\langle 1\rangle) \cup A_\psi\langle 0\rangle \\ 0, & \text{if } \mathrm{x} \in (\psi\langle 0\rangle \setminus A_\psi\langle 0\rangle) \cup A_\psi\langle 1\rangle \end{cases} \tag{34}$$

We are interested only on sets A that produce increasing operators. Such sets will be called *switching sets*.

In particular, if a switching set for ψ_{opt} is chosen in such a way to minimize the error increase $\Delta_M(\psi, \psi_{opt})$, then the resulting operator is an optimal increasing operator. Switching ψ_{opt} by a switching set A yields an increasing operator $\psi_{opt\,A}$ with error increase

$$\Delta_M(\psi_{opt\,A}, \psi_{opt}) = \sum_{\mathrm{x} \in A} c(\mathrm{x}) \tag{35}$$

From now on, we will consider the inversion set Q, switching costs $c(\mathrm{x})$, 1-set and 0-set relative to an

operator ψ_{opt}, without explicitly using the subscripts ψ_{opt}.

A *valid partition* of Q is an ordered pair (L, U) such that $L \cup U = Q$, $L \cap U = \emptyset$, and no element of L lies above any element of U (or equivalently, no element of U lies beneath any element of L). The sets L and U are called *lower* and *upper* sets of the partition, respectively.

The problem of finding a switching set that minimizes Eq. 35 can be modeled as a problem of finding a binary partition of Q. More specifically, let $A \subseteq Q$ be a switching set, and (L, U) be a valid partition of Q. If we let $L_A = (Q\langle 0\rangle \setminus A\langle 0\rangle) \cup A\langle 1\rangle$, $U_A = (Q\langle 1\rangle \setminus A\langle 1\rangle) \cup A\langle 0\rangle$, and $A_{(L,U)} = (L \cap Q\langle 1\rangle) \cup (U \cap Q\langle 0\rangle)$, then (L_A, U_A) is a valid partition of Q, $A_{(L,U)}$ is a switching set, $A_{(L_A,U_A)} = A$, and $(L_{A_{(L,U)}}, U_{A_{(L,U)}}) = (L, U)$, which means that switching sets are equivalent to valid partitions.

Because of this equivalence between switching sets and valid partitions, given a switching set A, the error increase of ${\psi_{opt}}_A$ (Eq. 35) can be rewritten in terms of the corresponding valid partition (L_A, U_A) as

$$\tilde{\Delta}(L_A, U_A) = \sum_{\mathrm{x} \in U_A\langle 0\rangle} c(\mathrm{x}) + \sum_{\mathrm{x} \in L_A\langle 1\rangle} c(\mathrm{x}) \qquad (36)$$

i.e., $\tilde{\Delta}(L_A, U_A) = \Delta_M({\psi_{opt}}_A, \psi_{opt})$. The valid partition of Q that minimizes $\tilde{\Delta}$ is an optimal partition of Q.

If just a small part of Q is analyzed at a time, it may be possible to decide in which side of the optimal partition it should belong. This idea is exploited by the switching algorithm to build the partition by progressively removing small subsets from the inversion set.

To simplify the presentation, for any $Z \subseteq Q$, let $\omega(Z) = \sum_{\mathrm{x} \in Z\langle 0\rangle} c(\mathrm{x}) - \sum_{\mathrm{x} \in Z\langle 1\rangle} c(\mathrm{x})$. *Feasible sets*, defined as follows, form a key element of the algorithm.

Definition 5.1 *Let $\mathcal{F}_{\mathcal{U}}$ be the class of non-empty subsets F of Q satisfying*
1. $(Q \setminus F, F)$ is a valid partition of Q, and
2. $\omega(F) < 0$.
A subset $F \in \mathcal{F}_{\mathcal{U}}$ is U-feasible *if and only if F is minimal in $\mathcal{F}_{\mathcal{U}}$ relative to $\subseteq$.*

Definition 5.2 *Let $\mathcal{F}_{\mathcal{L}}$ be the class of non-empty subsets F of Q satisfying*
1. $(F, Q \setminus F)$ is a valid partition of Q, and
2. $\omega(F) \geq 0$.
A subset $F \in \mathcal{F}_{\mathcal{L}}$ is L-feasible *if and only if F is minimal in $\mathcal{F}_{\mathcal{L}}$ relative to $\subseteq$.*

For convenience, $(\emptyset, \emptyset)$ is considered a valid and optimal partition of $\emptyset$. To refer to U- or L-feasible sets, without explicit distinction, we will simply use the term "feasible sets".

Theorem 5.1 *Let F be a feasible set of Q, and (L', U') be an optimal partition of $Q \setminus F$. Then,*
(a) if F is U-feasible then $(L', U' \cup F)$ is an optimal partition of Q, and
(b) if F is L-feasible then $(L' \cup F, U')$ is an optimal partition of Q.
(see [4] for a proof.)

It can be shown that an inversion set always contains at least one feasible set. The switching algorithm (Algorithm 5.1) starts with empty upper and lower sets, and then sequentially removes feasible sets from the remaining inversion set to one of the sides of the partition being built. Since it is a greedy algorithm, once a subset is removed, it will never be put back into Q.

Algorithm 5.1

1. Set $U \leftarrow \emptyset$ and $L \leftarrow \emptyset$.
2. If Q is empty, then return (L, U) and exit.
3. Search for a feasible set F in Q. If F is U-feasible, then do $U \leftarrow U \cup F$; if F is L-feasible, then do $L \leftarrow L \cup F$. Do $Q \leftarrow Q \setminus F$ and return to step 2.

More details and a proof that this algorithm works is given in [4]. A key point to understand this algorithm is to observe that when a feasible set is moved from the inversion set to one of the sides of the partition being built, the switchings implied by this moving do not affect other elements in the remaining inversion set.

An efficient implementation of the algorithm needs to find feasible sets quickly. Due to the minimality condition of feasible sets, the algorithm needs to make sure that no proper subset of the set to be tested is a feasible set. Hence, the search procedure may first check subsets with one minimal/maximal element that do not contain proper subsets not tested yet. In sequence, it may check subsets with two minimal/maximal elements that do not contain proper subsets not tested yet, and so on.

In general, feasible sets have few minimal / maximal elements and therefore the search need not go through all possibilities. However, it is possible that at some instance of the problem all feasible sets are very large. In such a case, the algorithm will need to test a large number of subsets before being able to find a feasible set, a critical situation in terms of processing time.

This situation can be avoided by using a relaxed definition of feasible sets, *m-feasible sets*, which allows

at most m minimal/maximal elements in the set.

If only m-feasible sets are searched, the search procedure described previously needs to consider only subsets with up to m minimal/maximal elements, and therefore the processing time can be controlled by changing the value of m. However, this relaxation may introduce suboptimality, meaning that the result may be non-optimal, although, conceptually, it is always possible to choose a sufficiently large m to produce an optimal result.

Due to the relaxation, there is no guarantee that a feasible set will be always found. Hence, step 3 of Algorithm 5.1 must be changed to treat cases where a feasible set is not found, as follows :

Algorithm 5.2

3′. Search for an m-feasible set F in Q.

- If F is found, and if it is U-m-feasible, then do $U \leftarrow U \cup F$; if F is L-m-feasible, then do $L \leftarrow L \cup F$. Do $Q \leftarrow Q \setminus F$.
- If F is not found, compute $\tilde{\Delta}(Q, \emptyset)$ and $\tilde{\Delta}(\emptyset, Q)$. If $\tilde{\Delta}(Q, \emptyset) \leq \tilde{\Delta}(\emptyset, Q)$ then do $L \leftarrow L \cup Q$, otherwise do $U \leftarrow U \cup Q$. Do $Q \leftarrow \emptyset$.

Return to step 2.

Once an optimal partition (L, U) is computed, the corresponding switching set $A_{(L,U)}$ can be computed. The optimal increasing set operator is the one whose basis consists of all minimal elements of ${\psi_{opt}}_A\langle 1\rangle$ (the 1-set of the resulting increasing operator.)

6 Designing Stack Filters

From the results of section 4 and 5, by computing the costs (Eq. 32) for the elements in the inversion set of ψ_{opt} (Eq. 30), the switching algorithm can be directly applied for the computation of a positive Boolean function corresponding to an optimal stack filter.

However, in practice, the probabilities needed for the computation of the costs are not available. Usually, they are estimated from sample images. Let

- N_i be the number of observations in the cross sections at level i
- $N_i(\mathrm{x})$ be the number of times x is observed in the cross sections at level i
- $N_i(\mathrm{x}, 1)$ be the number of times x is observed in the cross sections at level i with $y = 1$
- $N_i(\mathrm{x}, 0)$ be the number of times x is observed in the cross sections at level i with $y = 0$

The probabilities $P_i(\mathrm{x})$, $P_i(1|\mathrm{x})$ and $P_i(0|\mathrm{x})$ are usually estimated, respectively, by $\hat{P}_i(\mathrm{x}) = \frac{N_i(\mathrm{x})}{N_i}$, $\hat{P}_i(1|\mathrm{x}) = \frac{N_i(\mathrm{x},1)}{N_i(\mathrm{x})}$ and $\hat{P}_i(0|\mathrm{x}) = \frac{N_i(\mathrm{x},0)}{N_i(\mathrm{x})}$.

We show that, the estimation of the optimal operator and costs can be easily computed. Using the estimators above, an estimator $\hat{\psi}_{opt}$ for the optimal Boolean function is obtained by replacing the probabilities of Eq. 30 by the respective estimators. Letting $\hat{c}_1(\mathrm{x}) = \sum_{i=1}^{k} \hat{P}_i(\mathrm{x})\hat{P}_i(0|\mathrm{x})$ and $\hat{c}_0(\mathrm{x}) = \sum_{i=1}^{k} \hat{P}_i(\mathrm{x})\hat{P}_i(1|\mathrm{x})$, we obtain

$$\hat{\psi}_{opt}(\mathrm{x}) = \begin{cases} 1, & \text{if } \hat{c}_0(\mathrm{x}) > \hat{c}_1(\mathrm{x}), \\ 0, & \text{if } \hat{c}_0(\mathrm{x}) \leq \hat{c}_1(\mathrm{x}). \end{cases} . \tag{37}$$

Similarly, an estimator for the cost of Eq. 32 is given by

$$\hat{c}(\mathrm{x}) = \begin{cases} \hat{c}_1(\mathrm{x}) - \hat{c}_0(\mathrm{x}), & \text{if } \hat{\psi}_{opt}(\mathrm{x}) = 0, \\ \hat{c}_0(\mathrm{x}) - \hat{c}_1(\mathrm{x}), & \text{if } \hat{\psi}_{opt}(\mathrm{x}) = 1 \end{cases} \tag{38}$$

If $N_i = N$ for all $i \in K$, N being a positive integer (which is the case when we consider the cross sections of a gray-scale image), then both $\hat{\psi}_{opt}$ and $\hat{c}(\mathrm{x})$ can be simplified, respectively, to

$$\hat{\psi}_{opt}(\mathrm{x}) = \begin{cases} 1, & \text{if } \sum N_i(\mathrm{x}, 0) > \sum N_i(\mathrm{x}, 1), \\ 0, & \text{if } \sum N_i(\mathrm{x}, 0) \leq \sum N_i(\mathrm{x}, 1). \end{cases} \tag{39}$$

and

$$\hat{c}(\mathrm{x}) = \begin{cases} \frac{1}{N}[\sum N_i(\mathrm{x}, 0) - \sum N_i(\mathrm{x}, 1)], & \text{if } \hat{\psi}_{opt}(\mathrm{x}) = 0, \\ \frac{1}{N}[\sum N_i(\mathrm{x}, 1) - \sum N_i(\mathrm{x}, 0)], & \text{if } \hat{\psi}_{opt}(\mathrm{x}) = 1. \end{cases} \tag{40}$$

Hence, there is no need to estimate the probabilities for each of the cross sections.

7 Experimental Results

This section presents some applications of stack filters, designed using the switching algorithm proposed in section 5 and cost estimation discussed in section 6. After a filter Ψ_1 is designed from training images f and f_0 (where f is the image to be filtered and f_0 is the respective ideal image), a second filter Ψ_2 can be designed using $\Psi_1(f)$ and f_0 as the training images. This procedure can be repeated several times, that is, Ψ_n can be designed using $\Psi_{n-1}(\ldots(\Psi_1(f))\ldots)$ and f_0 as the training images. The filter Ψ_n is called an n-iteration filter. The experiments were performed on a dual Pentium Pro 200 MHz processor.

7.1 Impulse noise removal

The noise to be filtered consists of a composition of two-side impulse (uniformly distributed with probability of occurrence of 10%, and with amplitude 200) plus

horizontal dropout noise (i.e., horizontal line segments of intensity 255 with probability of occurrence 0.35%, whose length follows a normal distribution with mean 5 and variance 49). Figure 2 shows the observed-ideal pair of images used for the training of a stack filter over a 17-point window, and Fig. 3 shows an image corrupted by a realization of the same noise process (MAE=13.2344), and the respective filtered image by the designed filter ($MAE = 1.4053$). The training time was about 7 minutes.

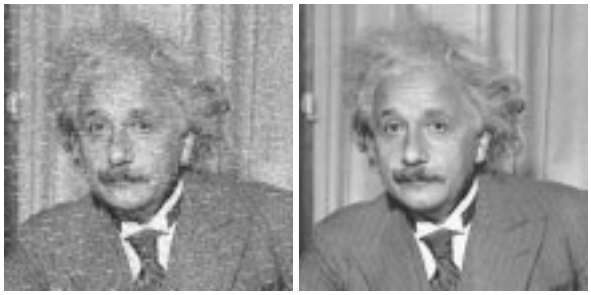

Figure 2: Training images.

Figure 3: Test and output images.

7.2 Speckle filtering

Speckle is a noise that appears in images obtained by coherent imaging systems such as the synthetic aperture radar (SAR) technologies [11, 12].

Here we show the use of stack filters for simulated speckle noise removing. Figure 4 shows, respectively, an image with 6 stripes with increasing gray-scale, a simulated 4-look amplitude SAR image, and the results obtained by a 1-iteration and 5-iteration stack filters over a 21-point window, trained using another simulation of the same type of noise. The MAE of the speckled image, 1-iteration and 5-iteration filtered images are, respectively, 16.752882, 4.9392, and 2.5917. The average speckle reduction index $E(\beta)$, where $\beta = \frac{\sqrt{Var(\Psi(f)(x))}}{E(\Psi(f)(x))}$, over the 6 homogeneous regions is 0.2622 for the original speckled image, 0.07968 for the 1-iteration and 0.04326 for the 5-iteration filtered images. The training time was about 2 hours for the first iteration, and a couple of minutes for the subsequent iterations.

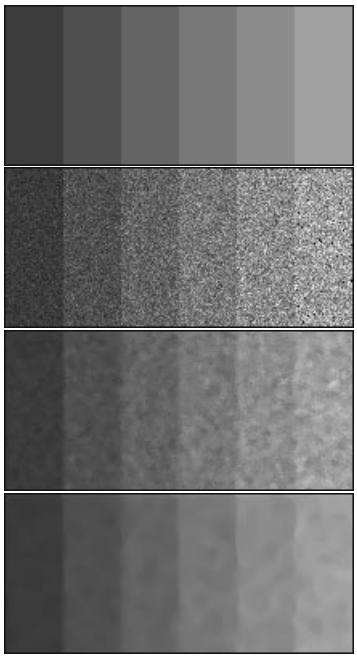

Figure 4: Ideal, test and output images.

Figure 5 shows, respectively, a synthetic image, a simulation of 1-look intensity SAR image, and the images produced by the 1 and 5-iteration stack filters obtained using another simulation of the same noise as the training image. The MAE of the speckled image, 1 and 5-iteration filtered images are, respectively, 21.643, 7.8531, and 3.2844. Here we considered the 17-point window. The training time was about 8 minutes.

8 Conclusion

This paper presented an overview of stack filters in the context of mathematical morphology and discrete images. The equivalence between stack filters and positive Boolean functions was studied. In particular, a lattice isomorphism between the set of stack filters and the set of positive Boolean functions was given, and the representation of stack filters as a union of erosions was recalled. The relationship between the design of optimal stack filters and the design of increasing set operators was studied. A previously proposed algorithm for the design of increasing operators for relatively large window size was shown to be directly applicable for the design of optimal MAE stack filters. Examples on

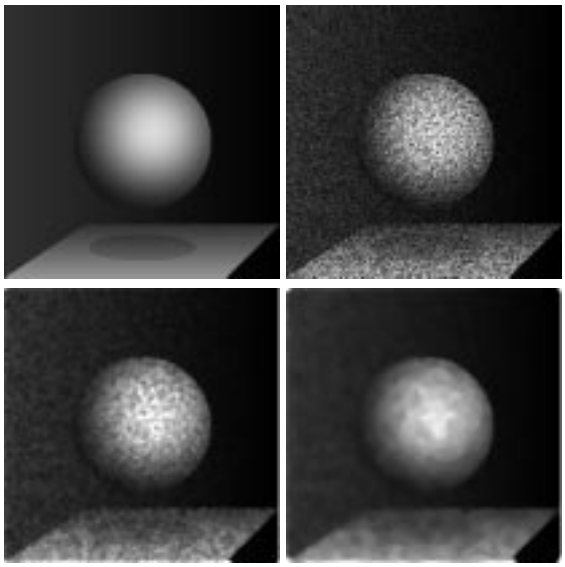

Figure 5: Ideal, test and output images.

impulse and speckle noise filtering illustrated the application of the design technique proposed. Next issues of this research include investigation of other error criterion and new applications of this design technique.

Acknowledgments

The authors thank Nelson D. A. Mascarenhas for the help received on several issues related to speckle noise filtering. N. S. T. Hirata acknowledges partial support from CNPq and FAPESP.

References

[1] J. Barrera, E. R. Dougherty, and N. S. Tomita. Automatic Programming of Binary Morphological Machines by Design of Statistically Optimal Operators in the Context of Computational Learning Theory. *Electronic Imaging*, 6(1):54–67, January 1997.

[2] J. Barrera, E. R. Dougherty, and N. S. T. Hirata. Design of Optimal Morphological Operators from Prior Filters. *Acta Steriologica*, 16(3):193–200, 1997. Special issue on Mathematical Morphology.

[3] E. R. Dougherty and R. P. Loce. Precision of Morphological-Representation Estimator for Translation-invariant Binary Filters: Increasing and Nonincreasing. *Signal Processing*, 40:129–154, 1994.

[4] N. S. T. Hirata, E. R. Dougherty, and J. Barrera. A Switching Algorithm for Design of Optimal Increasing Binary Filters Over Large Windows. *Pattern Recognition*, to appear, 1999.

[5] J. P. Fitch, E. J. Coyle, and N. C. Gallagher Jr. Median Filtering by Threshold Decomposition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32(6):1183–1188, December 1984.

[6] P. D. Wendt, E. J. Coyle, and N. C. Gallagher Jr. Stack Filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-34(4):898–911, August 1986.

[7] N. S. T. Hirata, J. Barrera, and E. R. Dougherty. Design of statistically optimal stack filters. Technical Report to appear, Instituto de Matemática e Estatística - USP, 1999.

[8] G. J. F. Banon. Characterization of Translation Invariant Elementary Operators for Gray-level Morphology. In E. R. Dougherty, F. Preteux, and S. Shen, editors, *Neural, Morphological, and Stochastic Methods in Image and Signal Processing*, SPIE Proceedings, pages 68–79, 1995.

[9] P. Maragos and R. W. Schafer. Morphological Filters: Part I: Their Set-Theoretic Analysis and Relations to Linear Shift-Invariant Filters. *IEEE Trans. Acoust. Speech Signal Process.*, ASSP-35:1153–1169, August 1987.

[10] E. J. Coyle and J.-H. Lin. Stack Filters and the Mean Absolute Error Criterion. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(8):1244–1254, August 1988.

[11] J. S. Lee, I. Jurkevich, P. Dewaele, P. Wambacq, and A. CosterLinck. Speckle Filtering of Synthetic Aperture Radar Images : A Review. *Remote Sensing Reviews*, 8:313–340, 1994.

[12] N. D. A. Mascarenhas. An Overview of Speckle Noise Filtering in SAR Images. In *Image Processing Techniques*, Proc. of First Latino-American Seminar on Radar Remote Sensing, pages 71–79, Buenos Aires, Argentina, 2-4 December 1996.