Journal of Mathematical Imaging and Vision, 2, 173–183 (1992).

Unification of Nonlinear Filtering in the Context of Binary Logical Calculus, Part I: Binary Filters

EDWARD R. DOUGHERTY
Center for Imaging Science, Rochester Institute of Technology, Rochester, NY 14623-0887

ROBERT M. HARALICK
Department of Electrical Engineering, University of Washington, Seattle, WA 98105

Abstract. The mathematical structure of nonlinear filtering is expressed in the context of binary logic. This first part of a two-part study concerns the binary setting. Operator properties, such as antiextensivity and idempotence, are expressed in finite logical expressions, as are the Matheron representation for morphological filters and its extension to translation-invariant operators, thereby giving simplicity to both operational properties and representations and also exposing the manner in which logic methods can be used for filter design and analysis. The second part of the study treats gray-scale filters.

Key words. nonlinear filter, morphological filter, image algebra, representation, cellular logic

1 Introduction

The present two-part paper seeks to address a basic question regarding mathematical imaging: What is an appropriate algebraic framework for image processing? The goal is not to seek the most abstract setting or the most complete, nor is it to present new mathematics or a new representation theory. In fact, in it we step back from some existing algebraic frameworks to reformulate some basic filter theory in a practical setting, that of binary logical calculus. The word *nonlinear* in the title reflects the fact that the filters considered herein are nonlinear, but given the limitation to practical cellular-logic implementation, it is in a sense redundant since there is no vector-space structure within which to ground linearity. Nevertheless, since we are concerned with algebraic structures classically identified with nonlinear image processing, the title appears appropriate.

Loosely, an *image algebra* is a collection of objects and operations between these objects that form an algebraic structure in which to formulate image-processing algorithms. Sternberg [1] has used the term *image algebra* to refer to morphological (Minkowski) algebra, and it is in this context that Crimmons and Brown [2] use the same term in relation to automatic shape recognition. More recently, *image algebra* has been used to describe two algebraic structures containing more structure than morphological algebra. One image algebra, developed by Ritter and colleagues [3], [4], is heterogeneous in that it contains many sorts of entities, the essential two being images and templates. The other, developed by Dougherty and Giardina [5], possesses a heterogeneous form and, as further developed by Dougherty [6], [7], a homogeneous form. Both image algebras serve to represent image-processing operations, both linear and nonlinear. In particular, they serve as a framework for linear operations because each contains the linear algebra of matrices as a subalgebra, and they serve as a framework for nonlinear operations because each possesses the necessary lattice structure. Regarding the necessity of structure within image algebra, Dougherty and Giardina [8] take special note of the induced nature of subalgebras. But what subalgebras need to be induced? Indeed, what subalgebras should be induced, given the computational nature of image processing?

In point of fact, digital images do not form a vector space relative to induced image addition and scalar multiplication. Strictly speaking, because the gray range is discrete and finite, image addition is not even closed. Even if we ignore the finitude of the gray range, we are still confronted by discreteness, so that the whole notion of linearity, including linear operators, cannot be subsumed within any image algebra that remains faithful to digital processing. This does not mean that richer mathematical structures cannot be of use, only that one should not see these as fully exhausting the algebraic question. It should be kept in mind that digital processing involves logic gates and bounded finite 0–1 representations. If we are to stick more closely to actual processing when we propose mathematical representations, then we need to stay within the confines of digital logic (or discrete set theory).

The set-theoretic properties of binary image filtering have been laid down by Matheron [9]. These include monotonicity, extensivity (antiextensivity), and idempotence. These are logical (or set-theoretic) concepts, and they play dominant roles in binary filtering. Matheron also carefully examines the role of translation invariance. This latter concept involves the translational structure within which image processing takes place and is related to image stationarity. If one reads Matheron closely, it is clear that the basic morphological operations of erosion, dilation, opening, and closing do not appear by chance in image processing. Indeed, he recognizes that any translation-invariant, monotonically increasing operator must be formed from a union of erosions and that any translation-invariant, increasing, antiextensive, and idempotent operator must be formed from a union of openings. Hence, by the very nature of digital image processing, mathematical morphology must play a key algebraic role, and therefore it is not surprising that Minkowski (morphological) algebra is central to the image algebras of both Ritter et al. and Dougherty and Giardina.

The algebraic framework for binary images established by Matheron [9] is extended to gray-scale images by means of lattice theory by Serra [10], [11] and Matheron [12]. They recognize that a complete lattice is the appropriate framework for the algebraic properties central to morphological processing and that the basic Matheron propositions thereby apply. Perhaps more importantly, the abstract lattice setting provides a framework for image processing that is more directly related to its logical (computable) nature than are richer algebras (containing vector-space subalgebras). More recent papers have further substantiated the proposition that a complete lattice provides the setting for the "algebraic basis of mathematical morphology" (Heijmans and Ronse [13], [14], Heijmans [15], Ronse [16]).

The central role of binary mathematical morphology arises from the set-theoretic aspects of binary processing; the key role of cellular logic arises from the manner in which image operators must be implemented on a digital computer. Cellular logic, and relevant related architectures, impress themselves on the algebraic analysis of image processing because processing is digital. It might be tempting to separate the computational and the abstract–mathematical problems, treating the former as architectural and the latter as algebraic; in fact, however, they are interrelated. Here is where we step back from the abstract set-theoretic analysis of Matheron and the subsequent lattice-theoretic approaches.

In part I of the present study we begin with cellular logic and explain the manner in which algebraic binary-filter theory emerges therefrom (see part II for gray-scale analysis). Such an approach naturally places those concepts typically considered to be morphological directly into the framework of cellular logic, which, of course, explains (in hindsight) the major role of cellular logic in the implementation of morphological processing. In particular, the minimal Matheron representation of increasing, translation-invariant binary-image operators as unions of erosions reduces to the well-known proposition that every finite positive Boolean expression possesses a minimal sum-of-products form. In addition, the extension of the Matheron representation by Banon and Berrera [17] to binary-image operators that are merely translation invariant, specifically, that these can be represented by unions of hit-or-miss operators, is seen to have a straightforward interpretation in cellular logic.

One might ask whether there is anything to be gained by the exercise of stepping back from the more general lattice framework, other than perhaps some readjustment of thinking. In fact, as will become evident, there is much more to be gained. By recognizing the practical Boolean nature of morphology and by formulating filter theory in the language of logic design, we see that standard computing tools, such as Karnaugh maps and Quine–McClusky reduction, can be applied to the construction of morphological operators. In operator design we are confronted by the need both to compose operator representations and, on the other hand, to decompose operators into constituent parts satisfying different algebraic constraints. Having a clear appreciation of the discrete logical character of nonlinear operators permits application of existing automatic routines.

2 Cellular-Logic Filters

We consider translation-invariant, moving-window operators on the space $\mathbf{S_B}$ of binary signals defined on Z, the set of integers. We assume that the window $W\langle m\rangle$ is centered at m and is of length $2M+1$. If Ψ is an operator of the specified type and $x = \{x[m]\}$ is a binary signal in $\mathbf{S_B}$, then

$$\Psi(x)[m] = \Psi(x[m-M], x[m-M+1], \ldots, x[m+M]), \qquad (1)$$

where we do not distinguish between the operator and the function rule defining the operator, calling them both Ψ. Relative to computer architecture, window logic is manifested as cellular logic, and therefore the filter Ψ will be called a *cellular-logic filter*. The choice of Z as the domain space for $\mathbf{S_B}$ is for convenience. Insofar as the subsequent logical analysis is concerned, the operative functional expression is (1), which depends only on denumerability (discreteness) of the domain and the finiteness of the window. In a setting different from Z (say, binary images defined on $Z \times Z$) the window can be of any shape and the ordering $x[m-M], x[m-M+1], \ldots, x[m+M]$ merely represents some given listing of the way in which the window is to be scanned. In particular, the assumption that the window is centered in (1) serves only the purpose of notational convenience.

Since Ψ is translation invariant, much of its analysis can be accomplished by considering the single output value $\Psi(x)[0]$, whose value depends on the window $W\langle 0\rangle = \{-M, -M+1, \ldots, M\}$ centered at the origin. $\Psi(x)[0]$ can be considered to be a binary functional on the set $\mathbf{M}$ of $\{0,1\}$-valued functions defined on $W\langle 0\rangle$. Every element of $\mathbf{M}$ can be represented by a string of ones and zeros, $(x[-M], x[-M+1], \ldots, x[M])$. From a set-theoretic perspective, every element of $\mathbf{M}$ is a subset of $W\langle 0\rangle$, where j lies in the subset if and only if $x[j] = 1$. Union and intersection in $W\langle 0\rangle$ are replaced by the logical maximum and minimum operations

$$x \vee y = (x[-M] \vee y[-M], \ldots, x[M] \vee y[M]), \qquad (2)$$

$$x \wedge y = (x[-M] \wedge y[-M], \ldots, x[M] \wedge y[M]) \qquad (3)$$

in $\mathbf{M}$. Moreover, the order relation $x \leq y$ if and only if $x[i] \leq y[i]$ for $i = -M, \ldots, M$ corresponds to the subset relation in $W\langle 0\rangle$.

As a binary functional, $\Psi(x)[0]$ can be written in logical format as a maximum of minima or, by using logical notation, as a canonical sum of products

$$\Psi(x)[0] = \sum x[-M]^{p[-M]} x[-M+1]^{p[-M+1]} \cdots x[M]^{p[M]}, \qquad (4)$$

where, for $j = -M, \ldots, M$, $p[j]$ is -1, 0, or 1 and where $x[j]^{-1}$ is the negation of $x[j]$ (also written $x[j]'$) and $x[j]^0$ means that the logical variable $x[j]$ does not appear in the product. In other words, $\Psi(x)[0]$ is a Boolean expression over $2M+1$ binary variables. As is well known, there are many expressions equivalent to (4), and, in fact, there are methods, such as

Karnaugh maps and the Quine–McClusky procedure, for minimizing the number of logic gates forming canonical sum-of-product expressions.

Owing to translation invariance the logical expression (4) applies to $\Psi(x)[m]$ for any m; the $p[j]$ remain the same:

$$\Psi(x)[m] = \sum x[m-M]^{p[-M]} \cdots x[m+j]^{p[j]} \cdots x[m+M]^{p[M]}. \qquad (5)$$

The variables $x[j]$ lie in the translated window $W\langle m\rangle = W\langle 0\rangle + m$ centered at m.

Another way of looking at the expansion (4) [and therefore at the expansion (5)] is to proceed in the following manner: (i) group the variables with $+1$ exponents in each product, and let $W_i\langle 0\rangle$ denote their product; (ii) group the variables with -1 exponents in each product, and let $W_i\langle 0\rangle'$ denote their product; (iii) ignore all variables with 0 exponent. Then, omitting null products, unless $\Psi(x)[0]$ is the zero function, expression (4) takes the form

$$\Psi(x)[0] = \sum W_i\langle 0\rangle W_i\langle 0\rangle', \qquad (6)$$

where it is possible for $W_i\langle 0\rangle$ or $W_i\langle 0\rangle'$ to be null, in which case it is denoted by 1.

Geometrically, $W_i\langle 0\rangle$ can be interpreted as a subwindow of $W\langle 0\rangle$ corresponding to positive Boolean variables (exponent $+1$) and $W_i\langle 0\rangle'$ can be interpreted as a subwindow corresponding to negative Boolean variables (exponent -1). We will subsequently make use of this convention by considering translates $W_i\langle m\rangle$ and $W_i\langle m\rangle'$. For instance, if $W\langle 0\rangle$ is the five-point window and

$$\Psi(x)[0] = x[-1]x[0]x[1] + x[0]x[1] + x[-1]x[1]'x[2]', \qquad (7)$$

then $W_1\langle 0\rangle = x[-1]x[0]x[1]$, $W_1\langle 0\rangle' = 1$, $W_2\langle 0\rangle = x[0]x[1]$, $W_2\langle 0\rangle' = 1$, $W_3\langle 0\rangle = x[-1]$, and $W_3\langle 0\rangle' = x[1]'x[2]'$. Among other things, the subwindow notation facilitates writing outputs at points other than the origin. Here, for instance, $\Psi(x)[m]$ is written simply as

$$\Psi(x)[m] = W_1\langle m\rangle + W_2\langle m\rangle + W_3\langle m\rangle W_3\langle m\rangle', \qquad (8)$$

where $W_i\langle m\rangle$ and $W_i\langle m\rangle'$ refer to the translated subwindows $W_i\langle 0\rangle + m$ and $W_i\langle 0\rangle' + m$, respectively.

3 Increasing Filters

A cellular-logic filter Ψ is *monotonically increasing* if $x \le y$ implies $\Psi(x) \le \Psi(y)$. Owing to translation invariance, Ψ is increasing if and only if $(x[-M], \ldots, x[M]) \le (y[-M], \ldots, y[M])$ implies $\Psi(x)[0] \le \Psi(y)[0]$. Ψ is increasing if and only if it can be expressed as a minimal sum of products for which there exists no negation in the expansion; i.e., $W_i\langle 0\rangle' = 1$ for all i. (In logical terminology, Ψ is a positive Boolean function.) Hence an increasing filter Ψ has a canonical representation

$$\Psi(x)[0] = W_1\langle 0\rangle + W_2\langle 0\rangle + \cdots + W_p\langle 0\rangle \qquad (9)$$

that possesses a minimal number of product terms. The minimal expression is unique and can be obtained from any other sum-of-products expression. In general, any number of products can be adjoined by summation to the minimal expression without changing the filter so long as each is formed from an existing product by adjoining positive factors. In the minimal expression (9) the factors of $W_i\langle 0\rangle$ do not form a subset of the factors of $W_j\langle 0\rangle$, for $j \ne i$.

There is a natural ordering on binary operators. Suppose Ψ_1 and Ψ_2 are two operators. We write $\Psi_1 \le \Psi_2$ if and only if $\Psi_1(x) \le \Psi_2(x)$ for any signal x. Now suppose Ψ_1 and Ψ_2 are increasing and in minimal sum-of-product form. Then $\Psi_1 \le \Psi_2$ if and only if for any product of Ψ_1 there exists a product of Ψ_2 whose factors form a subset of the factors of the given product for Ψ_1.

An increasing cellular-logic filter Ψ is said to be *antiextensive* [*extensive*] if $\Psi(x) \le x$ [$\Psi(x) \ge x$] for all x. Relative to a sum-of-products expression for $\Psi(x)[0]$, Ψ is antiextensive if and only if each product term of $\Psi(x)[0]$ contains $x[0]$. Ψ is extensive if and only if it possesses the singleton product term $x[0]$ in its minimal sum-of-products representation.

4 Iteration

Of great concern in filtering is iteration: given filters Ψ and Φ, what can be said about the product $\Phi\Psi$? For the moment, we consider arbitrary filters Ψ and Φ, not necessarily increasing, and we examine the sum-of-products representation for $\Phi\Psi$. The cumbersome part of the problem is this: when Φ operates on $\Psi(x)$, each variable $y[m]$ in $y = \Psi(x)$ is expressed as a sum-of-products of the original x variables lying in the window $W\langle m\rangle$ about $x[m]$. Thus the expression for $\Phi\Psi(x)[0]$ potentially includes the variables $x[-2M], x[-2M+1], \ldots, x[2M]$. The expression for $\Phi\Psi(x)[0]$ results from putting the expressions for $\Psi(x)[-M], \Psi(x)[-M+1], \ldots, \Psi(x)[M]$ into the expression for $\Phi(x)[0]$ in place of $x[-M], x[-M+1], \ldots, x[M]$, respectively. A key point is that once this has been done, reduction can be done to achieve a minimal-gate representation, and this can be accomplished automatically by some procedure such as the Quine–McClusky algorithm.

If Ψ and Φ happen to be increasing, the same reasoning applies; however, here reduction is much simpler. We need only expand the terms within the minimal sum-of-products representation for Φ when we replace the variables $x[j]$ by $\Psi(x)[j]$ and eliminate redundant products. This can always be done automatically.

As an illustration, consider the three-point window about the origin and let

$$\Psi(x)[0] = x[-1]x[0] + x[0]x[1]. \tag{10}$$

Then

$$\begin{aligned}\Psi\Psi(x)[0] &= (x[-2]x[-1] + x[-1]x[0])\\ &\quad (x[-1]x[0] + x[0]x[1])\\ &\quad +(x[-1]x[0] + x[0]x[1])\\ &\quad (x[0]x[1] + x[1]x[2])\\ &= \Psi(x)[0](x[-2]x[-1] + x[-1]x[0]\\ &\quad +x[0]x[1] + x[1]x[2])\\ &= \Psi(x)[0](x[-2]x[-1]\\ &\quad +\Psi(x)[0] + x[1]x[2])\\ &= \Psi(x)[0],\end{aligned} \tag{11}$$

the last equality following from the fact that for any logical expression ab, where $a \le b$, $ab = a$. For this particular example we obtain the very special relation $\Psi\Psi = \Psi$.

5 Idempotence

A filter Ψ is said to be *idempotent* if $\Psi\Psi = \Psi$. For increasing filters idempotence can be characterized in terms of sum-of-products expressions. Consider the minimal sum-of-products expression for an increasing filter Ψ. Some product terms of $\Psi(x)[0]$ contain $x[0]$, and some do not. Thus we can express $\Psi(x)[0]$ as

$$\begin{aligned}\Psi(x)[0] = x[0]\sum f_i(&x[-M], \ldots, x[-1],\\ &x[1], \ldots, x[M])\\ +\sum g_j(&x[-M], \ldots, x[-1],\\ &x[1], \ldots, x[M]),\end{aligned} \tag{12}$$

where f_i and g_j are products of the variables in the centered window $W\langle 0\rangle$, excluding the variable $x[0]$. If $x[0]$ happens to be a product term of $\Psi(x)[0]$, then one of the f_i is 1 and without loss of generality we assume $f_1 = 1$. If the second sum is empty, then Ψ is antiextensive; otherwise, it is not. If $f_1 = 1$, then Ψ is extensive; otherwise, it is not. We write the decomposition (12) as

$$\Psi(x)[0] = x[0]\Psi_0(x)[0] + \Psi_1(x)[0]. \tag{13}$$

Operating a second time by Ψ yields

$$\begin{aligned}\Psi\Psi(x)[0] = \Psi(x)[0]\sum f_i(&\Psi(x)[-M],\\ &\ldots, \Psi(x)[-1], \Psi(x)[1],\\ &\ldots, \Psi(x)[M]).\\ +\sum g_j(&\Psi(x)[-M],\\ &\ldots, \Psi(x)[-1], \Psi(x)[1],\\ &\ldots, \Psi(x)[M]).\end{aligned} \tag{14}$$

In terms of the decomposition (13) idempotence takes the form

$$\begin{aligned}\Psi(x)[0] = \Psi(x)[0]&\Psi_0(\Psi(x))[0]\\ &+\Psi_1(\Psi(x))[0],\end{aligned} \tag{15}$$

which is a logical identity of the form $a = ab + c$. A necessary condition for the identity is $c \le a$.

Two sufficient conditions are $c = a$ and $b \geq a \geq c$.

A key subcase concerning idempotence for an increasing filter Ψ is when the operator is antiextensive. In such a situation Ψ_1 is null, so that equation (15) is of the logical form $a = ab$, and hence a necessary and sufficient condition for idempotence is

$$\Psi(x)[0] \leq \Psi_0(\Psi(x))[0]. \quad (16)$$

This is precisely what happened for the antiextensive filter of equation (10). For it,

$$\Psi_0(x)[0] = x[-1] + x[1], \quad (17)$$
$$\Psi_0(\Psi(x))[0] = x[-2]x[-1] + x[-1]x[0] + x[0]x[1] + x[1]x[2]. \quad (18)$$

The filter of equation (10) belongs to the important subclass of all increasing antiextensive, idempotent cellular-logic filters.

The filters in this special class are called *τ-openings*, and within this class are the *openings*, which, in the context of a fixed window $W\langle 0\rangle$, will be called *$W\langle 0\rangle$-openings*. A $W\langle 0\rangle$-opening is defined by specifying a primitive product whose first factor is $x[0]$. To wit, let

$$h_0 = x[0]x[j_1]x[j_2]\cdots x[j_r], \quad (19)$$

where $0 < j_1 < \cdots < j_r \leq M$, be the primitive product. For $k = 1, 2, \ldots, r$ let

$$h_k = x[-j_k]x[j_1 - j_k]\cdots x[j_r - j_k]. \quad (20)$$

Define the $W\langle 0\rangle$-opening $\Psi(x)$ by

$$\Psi(x)[0] = h_0 + h_1 + \cdots + h_r. \quad (21)$$

Since $x[0]$ appears in every product, Ψ is antiextensive.

By using strictly logical calculus it can be shown that every $W\langle 0\rangle$-opening is idempotent. We consider $r = 2$, the proof for general r being similar but tedious. For $r = 2$ express h_0, h_1, and h_2, as $h_0 = x[0]x[p]x[q]$, $h_1 = x[-p]x[0]x[q-p]$, and $h_2 = x[-q]x[p-q]x[0]$, so that

$$\Psi(x)[0] = x[0](x[p]x[q] + x[-p]x[q-p] + x[-q]x[p-q]), \quad (22)$$

the sum being $\Psi_0(x)[0]$. Moreover,

$$\Psi_0(\Psi(x))[0] = \Psi(x)[p]\Psi(x)[q] + \Psi(x)[-p]\Psi(x)[q-p] + \Psi(x)[-q]\Psi(x)[p-q]. \quad (23)$$

There are three summands, each possessing two factors, forming $\Psi_0(\Psi(x))[0]$. The factors of the first summand are

$$\Psi(x)[p] = x[p](x[2p]x[p+q] + x[0]x[q] + x[p-q]x[2p-q]),$$
$$\Psi(x)[q] = x[q](x[p+q]x[2q] + x[q-p]x[2q-p] + x[0]x[p]). \quad (24)$$

Since each factor contains the summand $x[0]x[p]x[q] = h_0$, the product $\Psi(x)[p]\Psi(x)[q]$ also contains the summand h_0. The factors of the second summand forming $\Psi_0(\Psi(x))[0]$ are

$$\Psi(x)[-p] = x[-p](x[0]x[q-p] + x[-2p]x[q-2p] + x[-p-q]x[-q]),$$
$$\Psi(x)[q-p] = x[q-p](x[q]x[2q-p] + x[q-2p]x[2q-2p] + x[-p]x[0]). \quad (25)$$

Each factor contains the summand $x[-p]x[0]x[q-p] = h_1$. Hence $\Psi(x)[-p]\psi(x)[q-p]$ also contains the summand h_1. Finally, a similar computation shows that the final summand forming $\Psi_0(\Psi(x))[0]$ contains h_2, and therefore $\Psi_0(\Psi(x))[0] \geq \Psi(x)[0]$ and Ψ is idempotent.

Although every opening is a τ-opening, not every increasing, antiextensive, idempotent cellular-logic filter is an opening. For instance,

$$\Psi(x)[0] = x[0](x[-2] + x[-1] + x[1] + x[2]) \quad (26)$$

is increasing, antiextensive, and idempotent but is not an opening.

It is, however, a sum (union) of openings since

$$\Psi(x)[0] = (x[-1]x[0] + x[0]x[1]) + (x[-2]x[0] + x[0]x[2]) \quad (27)$$

and both summands are openings. The expression of τ-openings as unions of openings is a question that, starting with Matheron [9], has been long addressed in mathematical morphology.

6 Monotonic Cellular Logic and Binary Mathematical Morphology

The advantages of implementing binary mathematical morphology in cellular-logic architectures have long been recognized. The success of the cellular approach is based on the fact that binary morphological operations are actually reformulations of Boolean expressions, so that binary Minkowski (morphological) algebra is equivalent to cellular-logic algebra, which is itself simply Boolean algebra with translations. We examine this equivalence.

Suppose $\Psi(x)[0]$ is defined by a single product

$$\Psi(x)[0] = x[j_1]x[j_2]\cdots x[j_r], \qquad (28)$$

where $-M \leq j_1 < j_2 < \cdots < j_r \leq M$. Let $A_\Psi\langle 0\rangle = \{j_1, j_2, \ldots, j_r\}$ be the subset of $W\langle 0\rangle$ associated with the product $\Psi(x)[0]$. Then $\Psi(x)[0] = 1$ if and only if $A_\Psi\langle 0\rangle$ is a subset of the set corresponding to x, this latter set to be denoted by $\langle x\rangle$. In general, $\Psi(x)[m] = 1$ if an only if $A_\Psi\langle m\rangle$ is a subset of $\langle x\rangle$. Since $A_\Psi\langle m\rangle = A_\Psi\langle 0\rangle + m$, this equivalence can be expressed in morphological terms: if we let $\Psi^\wedge$ denote the set mapping corresponding to the logical mapping Ψ, then

$$\Psi^\wedge(\langle x\rangle) = \langle x\rangle \ominus A_\Psi\langle 0\rangle, \qquad (29)$$

where $\ominus$ denotes erosion. Because the collection of 0–1 signals is isomorphic to the collection of integer subsets, Ψ and $\Psi^\wedge$ are actually the same operator, so that (29) states that every single-product increasing logical binary operator defined over the window $W\langle 0\rangle$ is equivalent to an erosion whose structuring element lies in $W\langle 0\rangle$.

More generally, a cellular-logic operator Ψ is defined by a sum of products possessing no negations if and only if Ψ is monotonically increasing. Since the logical operation + is equivalent to union, Ψ is a positive Boolean expression if and only if it is equivalent to a union of erosions, the structuring elements in the erosion expansion corresponding to products in the logical expansion. In sum, we have four equivalent conditions: (i) Ψ can be expressed as a sum of products possessing no negations; (ii) Ψ is monotonically increasing as a logical operator; (iii) $\Psi^\wedge$ is monotonically increasing as a set operator; (iv) $\Psi^\wedge$ is a union of erosions.

Define the *kernel* of an increasing logical filter Ψ to be the collection Ker[Ψ] of all signals z for which $\Psi(z)[0] = 1$. Then $z \in$ Ker[Ψ] if and only if there is a product $x[j_1]\cdots x[j_r]$ in the sum-of-products expansion defining Ψ such that $z[j_1] = \cdots = z[j_r] = 1$, which is equivalent to saying that $A = \{j_1, \ldots, j_r\}$ is a subset of $\langle z\rangle$, which in turn means that 0 lies in $\langle z\rangle \ominus A$. Since A is one of the structuring elements forming the union of erosions comprising $\Psi^\wedge$, $0 \in \Psi^\wedge(\langle x\rangle)$. By definition, a set lies in the kernel of a set mapping if and only if the filtered version of the set contains the origin. Hence the kernel of Ψ as a logical operator is equivalent to the kernel of $\Psi^\wedge$ as a morphological filter.

If a set operator $\Psi^\wedge$ is increasing and translation invariant, the Matheron representation [9] states that $\Psi^\wedge$ is expressed as the union of erosions by kernel elements, namely,

$$\Psi^\wedge(S) = \cup\{S \ominus A : A \in \text{Ker}[\Psi^\wedge]\}. \qquad (30)$$

It was noticed by Maragos and Schafer [18] and by Dougherty and Giardina [19], [20] that the kernel expression is redundant. Bas[$\Psi^\wedge$] is called the *basis* for $\Psi^\wedge$ if (a) every element in the kernel possesses a subset in Bas[$\Psi^\wedge$] and (b) no two elements in Bas[$\Psi^\wedge$] are properly related by the subset relation. Bases are unique. If there exists a basis for $\Psi^\wedge$, then the kernel expansion of equation (30) can be replaced by an expansion over the basis of the filter. The defining conditions of a basis mean there is no redundancy in the Matheron representation.

A monotonically increasing cellular-logic operator Ψ possesses a minimal sum-of-products representation. In that minimal form no product is a proper subproduct of another product. But this says that no structuring element is a proper subset of another structuring element in the erosion expansion representing $\Psi^\wedge$, which is then precisely the basis form of the Matheron representation for $\Psi^\wedge$. Thus in the discrete-window context the Matheron basis representation of a translation-invariant, increasing set mapping is actually a restatement of the fact that every increasing logical operator over a finite set of variables has a minimal sum-of-products representation, the minimizing products being the filter basis.

As an illustration, consider the logical operator defined by

$$\Psi(x)[0] = x[-1]x[0](x[1] + x[2]) + x[-1](x[-2] + x[0]). \quad (31)$$

Logical calculus yields

$$\Psi(x) = x[-1]x[0]x[1] + x[-1]x[0]x[2] + x[-2]x[-1] + x[-1]x[0]. \quad (32)$$

Reduction yields the minimal sum-of-products representation

$$\Psi(x) = x[-1]x[0] + x[-2]x[-1]. \quad (33)$$

Direct translation of equation (32) yields an erosion representation of the set operator corresponding to Ψ, namely,

$$\Psi^\wedge(\langle x\rangle) = (\langle x\rangle \ominus \{-1,0,1\}) \cup(\langle x\rangle \ominus \{-1,0,2\}) \cup(\langle x\rangle \ominus \{-2,-1\}) \cup(\langle x\rangle \ominus \{-1,0\}). \quad (34)$$

Since $\{-1,0\}$ is a subset of both $\{-1,0,1\}$ and $\{-1,0,2\}$, the Matheron basis representation

$$\Psi^\wedge(\langle x\rangle) = (\langle x\rangle \ominus \{-1,0\}) \cup(\langle x\rangle \ominus \{-2,-1\}) \quad (35)$$

is obtained, and this representation corresponds to the minimal sum-of-products representation of equation (33).

In the context of the Matheron representation we see the morphological interpretation of openings. As defined by Matheron, an operator that is translation-invariant, increasing, antiextensive, and idempotent is called a τ-opening. The most basic τ-opening is the elementary opening defined as erosion followed by dilation with the same structuring element: for signal $\langle x\rangle$ and structuring element A the opening of $\langle x\rangle$ by A is defined by $\langle x\rangle \circ A = (\langle x\rangle \ominus A) \oplus A$. The morphological basis of the opening $\langle x\rangle \circ A$ consists of all translates of A that contain the origin. Consequently, if A is finite, $A = \{j_0, j_1, \ldots, j_r\}$, then

$$\langle x\rangle \circ A = \cup\{\langle x\rangle \ominus (A - j_k) : k = 0,1,\ldots,r\}. \quad (36)$$

Letting $h_0 = x[0]x[j_1 - j_0]\cdots x[j_r - j_0]$, we see that $\langle x\rangle \circ A$ is equivalent to $\Psi(x)$, where $\Psi(x)[0]$ is defined by equation (21). Hence, a cellular-logic opening (as we have defined it) is equivalent to a morphological opening. Regarding τ-openings, Matheron [9] has shown that an operator is a τ-opening if and only if it can be represented as a union of openings, and this is precisely the import of equation (27), which expresses the τ-opening Ψ defined in equation (26) as a sum (union) of openings.

A key advantage of the logical formulation of mathematical morphology is the ability to check properties and relationships automatically. For instance, since idempotence for binary morphological operators is equivalent to idempotence for logical operators and since the latter characterization is machine checkable, we ipso facto have machine algorithms to check the morphological property. A second important example concerns the Matheron representation. Given the Matheron representations of several filters, the Matheron representation of an iteration can be found by the same algorithm that reduces an iteration of sum-of-product expansions to a single minimal sum of products.

7 Cellular Logic and Hit-or-Miss Transformations

Positive Boolean expansions are equivalent to the Matheron representation; what about the general sum-of-products expression (4)? Let us again begin with a single product

$$\Psi(x)[0] = x[j_1]x[j_2] \cdots x[j_r]x[i_1]'x[i_2]' \cdots x[i_s]', \quad (37)$$

where $-M \leq j_1 < \cdots < j_r \leq M, -M \leq i_1 < \cdots < i_s \leq M$ and there does not exist a pair of indices j_a and i_b such that $j_a = i_b$. If we let $A = \{j_1, \ldots, j_r\}$ and $B = \{i_1, \ldots, i_s\}$, then $\Psi(x)[m] = 1$ if and only if $A + m$ is a subset of $\langle x \rangle$ and $B + m$ is a subset of $\langle x \rangle^c$, the complement of $\langle x \rangle$. This means that m lies in both $\langle x \rangle \ominus A$ and $\langle x \rangle^c \ominus B$. But the intersection of these two erosions is the hit-or-miss transform (Serra [21]) generated by the structuring pair (A, B):

$$\langle x \rangle \circledast (A, B) = (\langle x \rangle \ominus A) \cap (\langle x \rangle \ominus B), \quad (38)$$

and Ψ is equivalent to the hit-or-miss operator.

If we now consider the most general form of the sum-of-products Boolean expression in equation (4), we see that every translation-invariant, moving-window binary logical function is equivalent to a union of hit-or-miss operators with structuring elements in the window. Thus a general Boolean operator Ψ possesses a morphological equivalent $\Psi^\wedge$. In the discrete, moving window case, minimal-gate expressions can be found by considering the operator as a sum of products and applying some reduction algorithm.

As an illustration of how to employ the logic–morphology isomorphism, consider a four-point image window with the origin in the lower-left corner, so that $W\langle 0 \rangle = \{(0,0), (0,1), (1,0), (1,1)\}$ in the Cartesian grid. If we let x, y, z, and w denote the left–right, top-down raster scan of the four-point square, then every moving-window operator can be defined by a truth table consisting of strings of the form $xyzw$, where the operator Ψ takes the form $xyzw \to \Psi(xyzw)$. Suppose that we wish to find the minimal morphological implementation of Ψ, where Ψ is defined by the truth table output (in the usual order): 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1. A Karnaugh map reduction yields $\Psi(xyzw) = yx' + yz$. Its morphological equivalent is given by

$$\Psi^\wedge(S) = [S \circledast (A, B)] \cup [S \ominus C], \quad (39)$$

where $A = \{(1,1)\}$, $B = \{(0,1)\}$, and $C = \{(0,0), (1,1)\}$.

Just as the representation of a monotonically increasing cellular-logic operator as a minimal sum of products constitutes a finite logical realization of the Matheron erosion representation the interpretation of a general cellular-logic operator as a union of hit-or-miss transforms constitutes a finite logical realization of a different morphological representation theorem, namely, Banon and Berrera's extension [17] of the Matheron representation to translation-invariant set mappings that are not necessarily increasing. If $\Psi^\wedge$ is a translation-invariant set operator, then

$$\Psi^\wedge(S) = \cup\{S \otimes (A, B) : [A, B] < \mathrm{Ker}[\Psi^\wedge]\}, \quad (40)$$

where $[A, B] = \{T : A < T < B\}$, a subset of the power set, is called the *closed interval* with *extremities* A and B (Birkhoff [22]) and $S \otimes (A, B)$ is the hit-or-miss operator applied to S with the structuring pair (A, B^c). In certain circumstances, the representation (40) can be reduced. A closed interval in $\mathrm{Ker}[\Psi^\wedge]$ is said to be *maximal* if no other closed interval contained in $\mathrm{Ker}[\Psi^\wedge]$ properly contains it. The set $\mathbf{B}[\Psi^\wedge]$ of all maximal closed intervals in $\mathrm{Ker}[\Psi^\wedge]$ is called the *basis* of $\Psi^\wedge$, and $\mathbf{B}[\Psi^\wedge]$ is said to *satisfy the representation condition* for $\Psi^\wedge$ if and only if for any closed interval in $\mathrm{Ker}[\Psi^\wedge]$ there exists a closed interval in $\mathbf{B}[\Psi^\wedge]$ containing it. If the basis $\mathbf{B}[\Psi^\wedge]$ satisfies the representation condition, then the expansion (40) reduces to

$$\Psi^\wedge(S) = \cup\{S \otimes (A, B) : [A, B] \in \mathbf{B}[\Psi^\wedge]\}. \quad (41)$$

As with the Matheron representation for increasing cellular-logic operators, the representation (41) is related to a general cellular-logic operator Ψ by recognizing, as we have in equation (39), that Ψ corresponds to a set mapping $\Psi^\wedge$. To illustrate the relationship, we consider two examples over a three-point window.

First suppose

$$\Psi(x)[0] = x[-1]x[0]' + x[0]x[1]. \quad (42)$$

To lie in the kernel of Ψ, a signal z must be defined in one of the following four ways over $\{-1, 0, 1\}$:

$$z_1 : z_1[-1] = 1, z_1[0] = 0, z_1[1] = 0,$$
$$z_2 : z_2[-1] = 1, z_2[0] = 0, z_2[1] = 1,$$

$$z_3 : z_3[-1] = 0, z_3[0] = 1, z_3[1] = 1,$$
$$z_4 : z_4[-1] = 1, z_4[0] = 1, z_4[1] = 1. \quad (43)$$

Because Ψ operates only over a three-point window, values of z outside $\{-1, 0, 1\}$ play no role. Thus when applying expansion (41) we need consider only maximal closed intervals formed from the four three-point signals in (43). There are three of these: $[z_1, z_2]$, $[z_2, z_4]$, $[z_3, z_4]$. Thus representation (41) yields

$$\Psi(x)[0] = x[-1]x[0]' + x[-1]x[1] + x[0]x[1], \quad (44)$$

which is equivalent to the definition of Ψ in equation (42). Next consider

$$\Psi(x)[0] = x[-1]'x[0]'x[1]' + x[1]' + x[-1]x[0]x[1]. \quad (45)$$

To lie in Ker[Ψ] a signal z must be defined in one of the following five ways over $\{-1, 0, 1, \}$:

$$z_1 : z[-1] = 0, z_1[0] = 0, z_1[1] = 0,$$
$$z_2 : z[-1] = 1, z_2[0] = 0, z_2[1] = 0,$$
$$z_3 : z[-1] = 0, z_3[0] = 1, z_3[1] = 0,$$
$$z_4 : z[-1] = 1, z_4[0] = 1, z_4[1] = 0,$$
$$z_5 : z[-1] = 1, z_5[0] = 1, z_5[1] = 1. \quad (46)$$

There are only two maximal closed intervals formed from the five three-point signals of equation (46): $[z_1, z_4]$, $[z_4, z_5]$. Thus representation (41) yields

$$\Psi(x)[0] = x[1]' + x[-1]x[0], \quad (47)$$

which is equivalent to the original expression for Ψ in (45).

References

1. S. Sternberg, "Image algebra," unpublished notes, 1983.
2. T. Crimmons and W. Brown, "Image algebra and automatic shape recognition," *IEEE Trans. Aerospace Electron. Systems*, vol. 21, 1985.
3. G.X. Ritter and P.D. Gader, "Image algebra techniques for parallel image processing," *Parallel Distrib. Comput.*, vol. 4, 1987.
4. G.X. Ritter, J.N. Wilson, and J.L. Davidson, "Image algebra: an overview," *Comput. Vis., Graph., Image Process.*, vol. 49, 1990.
5. E.R. Dougherty and C.R. Giardina, *Mathematical Methods for Artificial Intelligence and Autonomous Systems,* Prentice-Hall: Englewood Cliffs, NJ, 1988. pp. 399–414
6. E.R. Dougherty, "A homogeneous unification of image algebra, part I: the homogeneous algebra," *Imaging Sci.*, vol. 33, 1989.
7. E.R. Dougherty, "A homogeneous unification of image algebra, part II: unification of image algebra," *Imaging Sci.*, vol. 33, 1989.
8. E.R. Dougherty and C.R. Giardina, "Image algebra—induced operators and induced subalgebras," *Proc. Soc. Photo-Opt. Instrum. Eng.*, vol. 845, 1987.
9. G. Matheron, *Random Sets and Integral Geometry*, John Wiley: New York, 1975.
10. J. Serra, "Mathematical morphology for complete lattices," in *Image Analysis and Mathematical Morphology*, vol. 2, J. Serra, ed., Academic Press: New York, 1988.
11. J. Serra, "Introduction to morphological filters," in *Image Analysis and Mathematical Morphology*, vol. 2, J. Serra, ed., Academic Press: New York, 1988.
12. G. Matheron, "Filters and lattices," in *Image Analysis and Mathematical Morphology*, vol. 2, J. Serra, ed., Academic Press: New York, 1988.
13. H. Heijmans and C. Ronse, "The algebraic basis of mathematical morphology, I: dilations and erosions," *Comput. Vis., Graph., Image Process.*, vol. 50, 1990.
14. H. Heijmans and C. Ronse, "The algebraic basis of mathematical morphology, II: openings and closings, *Comput. Vis., Graph., Image Process.*," vol. 54, 1990.
15. H. Heijmans, "Theoretical aspects of gray-level morphology," *IEEE Trans. Patt. Anal. Mach. Intell.* vol. 13, 1991.
16. C. Ronse, "Why mathematical morphology needs complete lattices," *Signal Process.*, vol. 21, 1990.
17. G.J. Banon and J. Berrera, "Minimal representations for translation invariant set mappings by mathematical morphology," *SIAM J. Appl. Math.*, vol. 51, 1991.
18. P. Maragos and R. Schafer, "Morphological filters—part I: their set-theoretic analysis and relations to linear shift-invariant filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, 1987.
19. E.R. Dougherty and C.R. Giardina, "A digital version of the Matheron representation theorem for increasing tau-mappings in terms of a basis for the kernel," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* July 1986.
20. C.R. Giardina and E.R. Dougherty, *Morphological Methods in Image and Signal Processing*, Prentice-Hall: Englewood Cliffs, NJ, 1988.
21. J. Serra, *Image Analysis and Mathematical Morphology,* Academic Press: New York, 1982.
22. G. Birkhoff, *Lattice Theory*, American Mathematical Society: Providence, RI, 1967.

Edward R. Dougherty is an associate professor at the Center for Imaging Science of the Rochester Institute of Technology and also serves as an industrial consultant. He holds an M.S. in computer science from Stevens Institute of Technology and a Ph.D. in mathematics from Rutgers University. He has written numerous papers in the area of mathematical morphology and image algebra and has authored or coauthored six books: *An Introduction to Morphological Image Processing, Matrix Structured Image Processing, Image Processing—Continuous to Discrete, Morphological Methods in Image and Signal Processing, Mathematical Methods for Artificial Intelligence and Autonomous Systems*, and *Probability and Statistics for the Engineering, Computing, and Physical Sciences.* He regularly teaches conference short courses in morphological image processing, including those for SPIE, and serves as a chair for two SPIE conferences, *Image Algebra and Morphological Image Processing* and *Nonlinear Image Processing.* His current research involves several principal areas: design of statistically optimal morphological filters, development of model-based approaches to morphological image analysis, granulometric characterization of texture, and development of intrinsically fuzzy approaches to mathematical morphology.

Robert M. Haralick is the Boeing Clairmont Egtvedt Professor in Electrical Engineering at the University of Washington. His recent work is in shape analysis and extraction using mathematical morphology, robust pose estimation, and techniques for making geometric inference from perspective projection information.

Professor Haralick has made a series of contributions in the high-level vision area, specifically dealing with the consistent labeling problem, arrangements, relation homomorphism, matching, and tree search. In the low-level and mid-level areas, Professor Haralick has worked in image texture analysis using spatial gray-tone co-occurrence texture features, and in the feature detection area he has developed the facet model for image processing.

Professor Haralick is a Fellow of IEEE for his contributions in computer vision and image processing. He serves on the Editorial Board of *IEEE Transactions on Pattern Analysis and Machine Intelligence.* He is the computer vision area editor for *Communications of the ACM.* He also serves as associate editor for *Computer Vision, Graphics, and Image Processing and Pattern Recognition.*

He received a B.A. degree in mathematics in 1964, a B.S. degree in electrical engineering in 1966, an M.S. degree in electrical engineering in 1967, and his Ph.D. in 1969, all from the University of Kansas.