Journal of Mathematical Imaging and Vision 22: 107–119, 2005


# Path Openings and Closings

HENK HEIJMANS*
*CWI, Amsterdam, The Netherlands*
henk.heijmans@cwi.nl

MICHAEL BUCKLEY AND HUGUES TALBOT†
*CSIRO Mathematical and Information Science, Locked Bag 17, North Ryde, NSW 1670 Australia*
michael.buckley@csiro.au
hugues.talbot@csiro.au

**Abstract.** This paper lays the theoretical foundations to path openings and closings.

The traditional morphological filter used for the analysis of linear structures in images is the union of openings (or the intersection of closings) by linear segments. However structures in images are rarely strictly straight, and as a result a more flexible approach is needed.

An extension to the idea of using straight line segments as structuring elements is to use constrained paths, i.e. discrete, one-pixel thick successions of pixels oriented in a particular direction, but in general forming curved lines rather than perfectly straight lines. However the number of such paths is prohibitive and the resulting algorithm by simple composition is inefficient.

In this paper we propose a way to compute openings and closings over large numbers of constrained, oriented paths in an efficient manner, suitable for building filters with applications to the analysis of oriented features, such as for example texture.

**Keywords:** oriented features, algebraic morphological filters, flexible linear morphological filters

## 1. Introduction

Practitioners of mathematical morphology are familiar with the importance of the structuring element in morphological and algebraic openings and closings. In spite of the infinite variety of available structuring elements, very few kinds of structuring elements are used in practice outside of a few specialized applications. The unit ball structuring elements of the discrete grid (e.g.: diamond, square and hexagon) define a first family of common structuring elements, useful for basic filtering, granulometries, etc.

Probably the second most used structuring element family is generated by some instance of the discrete line segment, which is used when linear and oriented structures are present in an application [8]. This limited choice can be at least partly blamed on the dearth of truly efficient algorithms for more arbitrary structuring elements [9].

However most structures in real-world images are not perfectly straight, and therefore using line segments as structuring elements in openings and closings can be inadequate in the common situation where there exist narrow, locally oriented features in an image of interest. In this case one might be interested in using structuring elements that are themselves narrow and oriented, but not perfectly straight. Unfortunately generating useful morphological filters in the usual way

*This paper was written at CSIRO MIS while on leave from CWI.
†Author to whom all correspondence should be addressed.

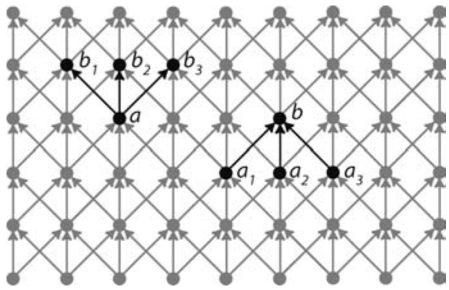



*Figure 1.* $b_1$, $b_2$, $b_3$ are successors of *a* and $a_1$, $a_2$, $a_3$ are the predecessors of *b*.

by composition leads to computationally expensive and impractical algorithms.

An earlier method for path-based filtering using morphological ideas was devised by Vincent [10] and provided the inspiration for the early work on path openings and closings. However Vincent's filter is based on local *sums* of pixels values whereas the path openings and closings described in this paper are based on local *minima* and *maxima*. Vincent's filter therefore behaves more like an oriented smoothing, and does not constitute a morphological filter.

In this paper we introduce the concept of path openings and closings, i.e. morphological filters that use families of structuring elements consisting of variously constrained paths, for which there exists algorithms as efficient as those using the usual families of straight line segments.

Path openings were originally proposed in [1] in an algorithmic, practical but theoretically incomplete manner. Here we are more concerned with laying down the theoretical foundations of these useful filters.

## 2. Adjacencies, Dilations and Paths

### *2.1. Adjacencies*

Let $E$ be a given set of points representing pixel locations. Define a *directed graph* on these points via a binary adjacency relation '$\mapsto$'. Specifically, $x \mapsto y$ means that that there is an edge going from $x$ to $y$. If $x \mapsto y$, we call $y$ a *successor* of $x$ and $x$ a *predecessor* of $y$. These concepts are illustrated in Fig. 1. Here $b_1, b_2, b_3$ are successors of $a$ and $a_1, a_2, a_3$ are the predecessors of $b$.

The relation '$\mapsto$' is, in general, neither reflexive nor symmetric.[1] We show some examples in Fig. 2.

Note that in the first three examples in Fig. 2, the adjacency relation is periodic away from the borders. The fact that we can choose any adjacency as a starting point enables us to handle the border in a consistent and flexible manner. Note also the following major difference between the adjacency relation in Fig. 2(a) and (c) and the one in (b). In (a) and (c), the graph structure is translation invariant with respect to any translation (again away from the borders), whereas in (b), this is only true if translation takes place over an even number of rows. Rephrased in terms of the dilation (see next subsection), this means that the structuring element is different at odd and at even rows. We will briefly address the issue of choosing the adjacency relation in Section 9.

### *2.2. Dilations*

Using the adjacency relation we have, for each point $x$, a set of its successors with respect to '$\mapsto$'. Denote this by $\delta(\{x\})$. That is,

$$\delta(\{x\}) = \{y \in E : x \mapsto y\}.$$

This can be generalised to arbitrary subsets $X$ of $E$ as follows:

$$\delta(X) = \{y \in E : x \mapsto y \text{ for some } x \in X\}.$$

In other words, $\delta(X)$ comprises all points which have a predecessor in $X$. In similar fashion we define the set $\check{\delta}(X)$ as the set of points which have a *successor* in $X$.

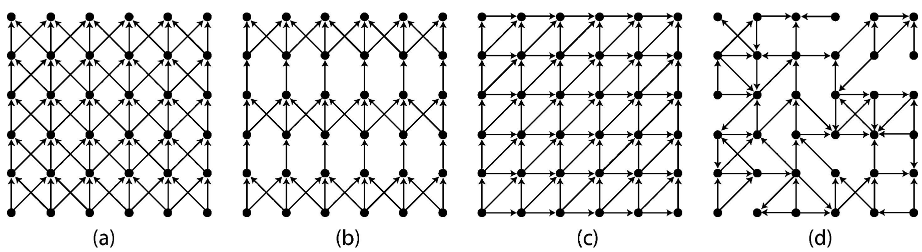



*Figure 2.* Directed graphs.

The operators $\delta$ and $\breve{\delta}$ have the property that $\delta(\cup_i X_i) = \cup_i \delta(X_i)$ for any sets $X_i$. This is the defining algebraic property of a *dilation*. Hence we refer to these operators as dilations.

Here we have first provided the adjacency relation '$\mapsto$' and defined the dilation in terms of '$\mapsto$'. It is straightforward to see that we might as well start with any dilation $\delta$ on $\mathcal{P}(E)$ and define the adjacency relation as

$$x \mapsto y \quad \text{if } y \in \delta(\{x\}).$$

Both approaches are equivalent and it is a matter of taste which one is taken.

### *2.3. Paths*

The $L$-tuple $\boldsymbol{a} = (a_1, a_2, \ldots, a_L)$ is called a *path of length L* if $a_k \mapsto a_{k+1}$, or equivalently, if

$$a_{k+1} \in \delta(\{a_k\}), \quad \text{for } k = 1, 2, \ldots, L-1.$$

Henceforth we refer to such a path as a $\delta$-path of length $L$. It is evident that $\boldsymbol{a} = (a_1, a_2, \ldots, a_L)$ is $\delta$-path if and only if the reverse path $\breve{\boldsymbol{a}} = (a_L, a_{L-1}, \ldots, a_1)$ is a $\breve{\delta}$-path and obviously, both paths have the same length $L$. We denote the set of all $\delta$-paths of length $L$ by $\Pi_L$ and the set of all $\breve{\delta}$-paths of length $L$ by $\breve{\Pi}_L$. Given a path $\boldsymbol{a}$ in $E$, we denote by $\sigma(\boldsymbol{a})$ the set of its elements:

$$\sigma(a_1, a_2, \ldots, a_L) = \{a_1, a_2, \ldots, a_L\}.$$

The set of $\delta$-paths of length $L$ contained in a subset $X$ of $E$ is denoted by $\Pi_L(X)$, i.e.,

$$\Pi_L(X) = \{a \in \Pi_L : \sigma(a) \subseteq X\}$$

and the $\breve{\delta}$-paths of length $L$ in $X$ by $\breve{\Pi}_L(X)$.

## 3. Path Opening

We define the set $a_L(X)$ as the union of all $\delta$-paths of length $L$ contained in $X$:

$$\alpha_L(X) = \bigcup\{\sigma(\boldsymbol{a}) : \boldsymbol{a} \in \Pi_L(X)\}.$$

It is not difficult to establish that the operator $\alpha_L$ has the algebraic properties of an *opening*, specifically increasingness, anti-extensivity and idempotence, and we call it the *path opening*.

Moreover in the cases of the usual periodic adjacencies, $\alpha_L$ is the supremum of morphological openings by a certain class of structuring elements, and the number of structuring elements in this class grows exponentially with $L$. For example, for an unbounded domain and the adjacency shown in Fig. 1(a), there are $3^{L-1}$ paths of length $L$ beginning from any point. None of these is a translation of another, and hence as structuring elements for morphological openings or closings they are distinct. The path opening $a_L$ can be shown to be the supremum of morphological openings by all of these $3^{L-1}$ structuring elements. In Section 4 we demonstrate how this opening may be computed with cost which grows linearly with $L$, even though the number of structuring elements implicitly involved grows exponentially with $L$. This is analogous to dynamic programming algorithms for shortest-paths in which the minimal path from an exponential collection of paths is computed in linear time.

We can define the reciprocal path opening $\breve{\alpha}_L(X)$ in a similar way. Since $\boldsymbol{a} \in \Pi_L(X)$ iff $\breve{\boldsymbol{a}} \in \breve{\Pi}_L(X)$ and $\sigma(\boldsymbol{a}) = \sigma(\breve{\boldsymbol{a}})$ we get immediately that

$$\alpha_L = \breve{\alpha}_L.$$

It is obvious that $\alpha_1 = \mathsf{id}$. We can show that that

$$\alpha_{L+1} \le \alpha_L \quad \text{for} \quad L \ge 1.$$

To prove this, assume that $x \in \alpha_{L+1}(X)$. Thus there is a $\delta$-path $(a_1, a_2, \ldots, \alpha_{L+1})$ of length $L+1$ which contains $x$ and lies inside $X$. But then both $\delta$-paths $(a_1, a_2, \ldots, a_L)$ and $(a_2, a_3, \ldots, a_{L+1})$ of length $L$ lie inside $X$ and at least one of them must contain the point $x$. This proves that $x \in \alpha_L(X)$, too. In Fig. 3 we show an example of a path opening where $L = 6$.

## 4. Computation of the Path Opening

In this section we define "first-point sets", $\psi_k(X)$ and derive relationships between these and the opened sets $\alpha_L(X)$ which allow feasible computation of path openings.

### *4.1. Path Decomposition*

By definition, $x \in \alpha_L(X)$ iff there exists a $\delta$-path $\boldsymbol{a} \in \Pi_L(X)$ that contains $x$, i.e., $x = a_k$ for some $k$ between

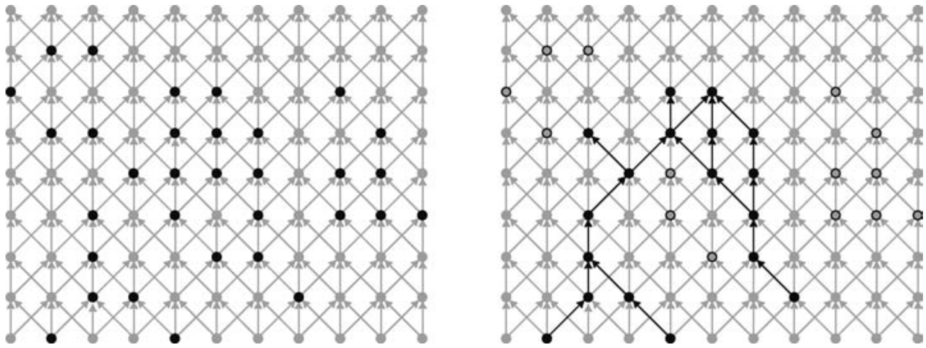


*Figure 3.* A set $X \subseteq E$ (black points at the left) and its opening $a_6(X)$ (black points at the right). Unfilled points at the right have been discarded. Adjacency graph is in light grey and identical to that of Fig. 2a.

1 and $L$. In that case we have

$$(a_1, a_2, \ldots, a_{k-1}, x) \in \Pi_k(X) \quad \text{and} \\ (x, a_{k+1}, \ldots, a_L) \in \Pi_{L-k+1}(X). \tag{1}$$

The first condition can be rewritten as

$$(x, a_{k-1}, a_{k-2}, \ldots, a_1) \in \breve{\Pi}_k(X). \tag{2}$$

We define the operator $\psi_k$ as

$$\psi_k(X) = \{a_1 : \boldsymbol{a} \in \Pi_k(X)\},$$

that is, $\psi_k(X)$ contains the first point of every $\delta$-path of length $k$ in $X$. The operator $\breve{\psi}_k$ is defined analogously. Obviously, $\psi_1 = \breve{\psi}_1 = \mathsf{id}$.

Now the first condition in (1), which is equivalent to (2), can be written as $x \in \breve{\psi}_k(X)$, and the second condition can be written as $x \in \psi_{L-k+1}(X)$. Combined, they give

$$x \in \breve{\psi}_k(X) \cap \psi_{L-k+1}(X) \text{ for some } k.$$

By taking the supremum over all the $k$, this is equivalent to

$$\alpha_L(X) \subseteq \bigcup_{k=1}^{L} (\breve{\psi}_k(X) \cap \psi_{L-k+1}(X))$$

By definition of $\breve{\psi}_k$ and $\psi_k$, the union of all of these contains the union of all the paths of length $L$ contained in $X$. By definition of $\alpha_L$ we have:

$$\bigcup_{k=1}^{L} (\breve{\psi}_k(X) \cap \psi_{L-k+1}(X)) \subseteq \alpha_L(X).$$

Using the complete lattice notation, we have shown that

$$a_L = \bigvee_{k=1}^{L} (\breve{\psi}_k \wedge \psi_{L-k+1}). \tag{3}$$

Note that $a_2 = \mathsf{id} \wedge (\delta \vee \breve{\delta})$, which is known in the literature as the *annular opening* [2,6].

Finally we note that the semi-group property

$$\psi_k \psi_l = \psi_{k+l-1}$$

holds for all $k, l \geq 1$. The proof of this is straightforward.

### 4.2. *Recursive Structure of the Operators $\psi_k$*

We will prove below that the following relations hold:

$$\psi_{k+1} = \mathrm{id} \wedge \breve{\delta}\psi_k \quad \text{and} \quad \breve{\psi}_{k+1} = \mathrm{id} \wedge \delta\breve{\psi}_k\,. \tag{4}$$

The decomposition of $a_L$ in (3) together with the iterative formulas in (4) provide an efficient algorithm for the computation of the path opening $a_L$. As shown in Algorithm 1, we compute the path opening $\alpha_L(X)$ given the set $X$ and dilation operators $\delta$ and $\breve{\delta}$. If we have a finite domain $E$ of size $N$ then unions and intersections can be computed in $O(N)$ operations. For simple periodic adjacencies such as those in Fig. 2(a)–(c), the dilations $\delta$ and $\breve{\delta}$ can also be computed in $O(N)$ operations. This gives a total cost for Algorithm 1 of $O(LN)$ for both operations and memory.

**Algorithm 1** (Binary Path-Opening).

// Recursive computation of $Y_k = \psi_k(X)$ and
// $\breve{Y}_k = \breve{\psi}_k(X)$ using Eq. (4)
$Y_1 = \breve{Y}_1 = X$
**for** $k = 1$ **to** $L - 1$
    $Y_{k+1} = X \cap \breve{\delta}(Y_k)$
    $\breve{Y}_{k+1} = X \cap \delta(\breve{Y}_k)$
**end**
// Construction of opening $A_L = \alpha_L(X)$ using Eq. (3)
$A_L = \emptyset$
**for** $k = 1$ **to** $L$
    $A_L = A_L \cup (Y_k \cap \breve{Y}_{L-1+k})$
**end**

We will prove only the first identity in (4) as the second is nothing but its reciprocal version. To prove '$\leq$' assume that $x \in \psi_{k+1}(X)$. This means that there exist $a_2, \ldots, a_{k+1}$ such that $(x, a_2, \ldots, a_{k+1}) \in \Pi_{k+1}(X)$. Now $(a_2, \ldots, a_{k+1}) \in \Pi_k(X)$ and $x \in \breve{\delta}(\{a_2\})$. Since $a_2 \in \psi_k(X)$ this yields that $x \in \breve{\delta}(\psi_k(X))$, and we conclude that $x \in X \cap \breve{\delta}(\psi_k(X))$.

To prove '$\geq$', let $x \in (\mathsf{id} \wedge \breve{\delta}\psi_k)(X)$, i.e., $x \in X$ and $x \in \breve{\delta}(\{y\})$ with $y \in \psi_k(X)$. The latter means that there exist $a_2, \ldots, a_k$ such that $(y, a_2, \ldots, a_k) \in \Pi_k(X)$. Now $(x, y, a_2, \ldots, a_k) \in \Pi_{k+1}(X)$, which yields that $x \in \psi_{k+1}(X)$.

The path opening $\alpha_L$ depends strongly upon the dilation, or equivalently, the adjacency relation. This is clearly seen in Fig. 4 where we have computed the opening $\alpha_5(X)$ of a set $X$ for three different adjacencies. A union of openings is an opening [5], therefore we can, for example, take the union of the first two openings in Fig. 4, i.e., two figures in the middle, to get an opening that allows both horizontal and vertical oriented paths. Note however, that this is not the same as combining both adjacencies into one and computing the opening with respect to this new adjacency.

## 5. Opening Transform

Often, we are interested in all openings $\alpha_L(X)$ of a set $X$ for a range of values of $L$ rather than for a single value only. For example, it is quite common that we do not know beforehand which $L$ to choose in a particular application. In such cases it may be more efficient to compute the so-called *opening transform* of the image. Given a set $X \subseteq E$ and an ordered family of openings $\mathcal{A} = \{\alpha_L\}$, the opening transform $A_X$ of $X$ with respect to $\mathcal{A}$ is a function mapping the domain $E$ into $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ such that its threshold sets correspond with the various openings $\alpha_L(X)$: see (6) below for an exact formulation.

Throughout this section we assume that there exist only finite acyclic paths. More precisely we assume that there exists an integer $N \geq 1$ such that $\delta^N(\{x\}) = \emptyset$, for every $x \in E$, i.e., $\delta^N(E) = \emptyset$. Note, however, that this assumption does not necessarily mean that $E$ is finite. Furthermore, we take $N$ to be the smallest integer with this property. Thus the maximal length of a path in $E$ is $N$. Define $\lambda(x)$ as the maximal length of a $\delta$-path with begin-point $x$:

$$\begin{aligned}\lambda(x) &= \max\{L \geq 1 : \exists\, a \in \Pi_L \text{ such that } a_1 = x\} \\ &= \max\{L \geq 1 : x \in \breve{\delta}^{L-1}(E)\}.\end{aligned}$$

Obviously, if $x \mapsto y$ then $\lambda(x) \geq \lambda(y) + 1$. Moreover, it is not difficult to prove that

$$\lambda(x) = 1 + \max\{\lambda(y) : x \mapsto y\},$$

where the maximum is taken to be zero if $x$ has no successors. Similarly $\breve{\lambda}(x)$ is the maximal length of a path with endpoint $x$. Then

$$\Lambda(x) = \lambda(x) + \breve{\lambda}(x) - 1, \tag{5}$$

is the length of the longest path that contains $x$. This is a direct consequence of Eqs. (1) and (2).

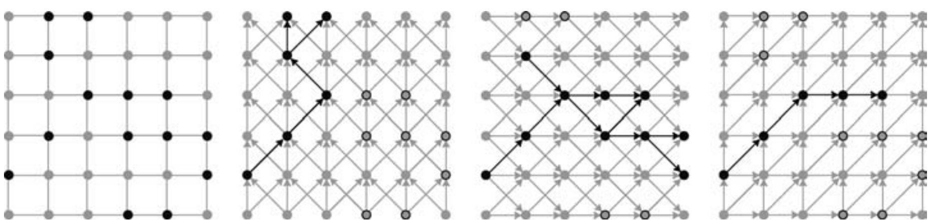


*Figure 4.* A set $X \subseteq E$ (left) and its opening $\alpha_5(X)$ for three different adjacencies.

Define the disjoint partition $E_1, E_2, \ldots, E_N$ of $E$ by

$$E_L = \{x \in E : \lambda(x) = L\}.$$

Similarly $\breve{E}_L$ is defined with $\breve{\lambda}$ instead of $\lambda$. In the figures at the left in Fig. 5 we depict this partition for two different adjacency relations; here the arrows indicate the relation $x \mapsto y$. Note that $E_L$ and $\breve{E}_L$ can be computed easily via a distance transform algorithm.

Now we define a function $F_X : E \to \mathbb{Z}_+$ by means of Algorithm 2 which resembles a geodesic propagation algorithm.

**Algorithm 2.**

$F_X = 0$ on $E$ // Initialisation
**for** $k = 1$ **to** $N$
  **for** $x \in E_k \cap X$
    $F_X(x) = 1 + \max\{F_X(y) \mid y \mapsto x\}$
  **end**
**end**

The second column of Fig. 5 shows the function $F_X$ for a given image $X$ (grey pixels). In a similar way we can define $\breve{F}_X$ by using the partition $\breve{E}_1, \breve{E}_2, \ldots, \breve{E}_N$.

The following lemma shows that $\psi_k(X)$ can be obtained by thresholding of $F_X$.

**Lemma 5.1.** *With the definitions given before we have*

$$\psi_k(X) = \{x \in E : F_X(x) \geq k\} \quad \textit{and}$$
$$\breve{\psi}_k(X) = \{x \in E : \breve{F}_X(x) \geq k\},$$

*for* $k = 1, 2, \ldots, N$.

**Proof:** Let $x \in E_k$ and suppose that $F_X(x) = l$. Obviously, $l \leq k$ and there must exist a $\delta$-path $x = a_1, a_2, \ldots, a_l$ in $\Pi_l(X)$ such that $a_i \in E_{k-i+1}$ and $F_X(a_i) = l - i + 1$. This implies that $x \in \psi_l(X)$. Conversely, if $x \in \psi_l(X)$, then there is a path $x = a_1, a_2, \ldots, a_l$ in $\Pi_l(X)$. Now if $x \in E_k$, where $k \geq l$, then $a_i \in E_{k-i+1}$ and $F_X(a_i) = l - i + 1$. □

This lemma can be used to prove the following result.

**Proposition 5.2.** *The function* $A_X = F_X + \breve{F}_X - 1$ *is the opening transform of* $X$, *that is*

$$\alpha_L(X) = \{x \in E : A_X(x) \geq L\}, \tag{6}$$

*for every* $L \geq 1$ *and* $X \subseteq E$.

**Proof:** We use the expression for $\alpha_L$ in (3) which says that $x \in \alpha_L(X)$ implies that $x \in \psi_k(X) \cap \breve{\psi}_{L-k+1}(X)$ for some $k = 1, 2, \ldots, L$. Therefore, $F_X(x) \geq k$ and $\breve{F}_X(x) \geq L - k + 1$, which yields that $A_X(x) \geq L$.

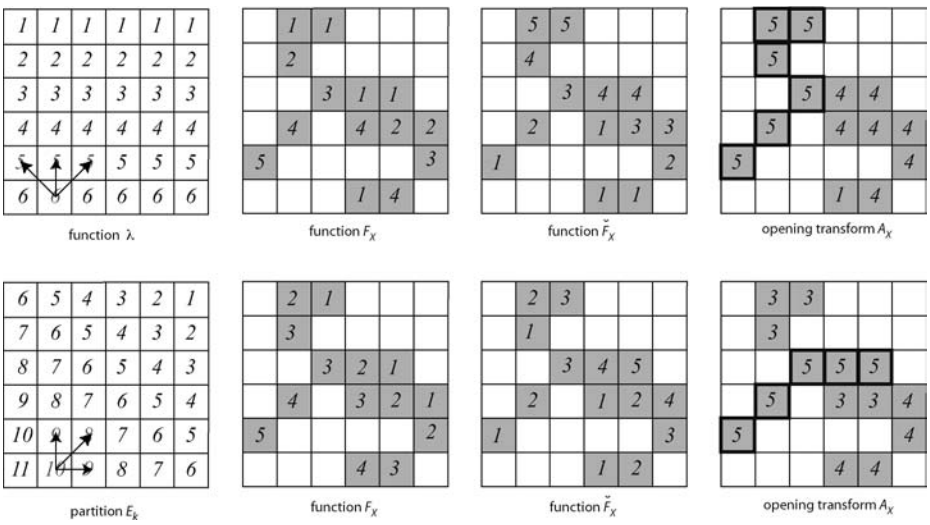



*Figure 5.* Computation of the opening transform for two different adjacencies indicated by the arrows in the first column, which also shows the partition of $E$. The second and third columns depict the functions $F_X$ and $\breve{F}_X$ for the set $X$ represented by the grey pixels. The right column shows the opening transform $A_X$ and the opened set $\alpha_5(X)$ represented by the pixels with the thick boundaries.

This proves '$\subseteq$' in (6). The converse is proved similarly. $\square$

In Fig. 5, the algorithm for the opening transform is shown for two different adjacencies, namely $(a)$ and $(c)$ in Fig. 2.

## 6. Border Issues

Throughout this section, $E$ will be a finite rectangular window within $\mathbb{Z}^2$, and the adjacency on $E$ is the restriction of a periodic adjacency on $\mathbb{Z}^2$ like in Fig. 2 (a)–(c). The *inward boundary* of $E$, denoted by $\partial E$, is the set of points in $E$ which have a predecessor outside $E$:

$$\partial E = \{x \in E : \exists\, y \in E^c \text{ such that } y \mapsto x\},$$

where $E^c$ is the complement of $E$. The *outward boundary* of $E$, denoted by $\breve{\partial} E$, is given by

$$\breve{\partial} E = \{x \in E : \exists\, y \in E^c \text{ such that } x \mapsto y\},$$

There are various ways to deal with the border problem:

(a) We can simply ignore the existence of the borders and treat paths that contain boundary points in the same way as any other path. In fact, this is the choice that we have implicitly made so far.
(b) The other extreme is to set the length of a path that crosses the border to $+\infty$, meaning that all points on such a path are contained in every opening $\alpha_L(X)$. In fact, this choice means that we extend $X$ outside $E$ by adding all points in $E^c$.
(c) An intermediate option is to try to compensate for the points cut off by restricting to a finite window: we replace the computed length $L$ of a path that has a begin-point in $\partial E$ or an endpoint in $\breve{\partial} E$ by $h(L)$. One possible choice for $h$ would be $h(L) = 2L$. Such a choice could be justified by the presumption that on average only half of the path falls inside the window. Another possibility is to add a fixed compensation to the length of a border-crossing path, i.e., $h(L) = L + L_0$. Note, that one might use different compensation functions for paths that start on $\partial E$ and end on $\breve{\partial} E$.
(d) A possibility which is easy to implement is to enlarge the window $E$ with a border $B$ of thickness $L_0$. Denote by $\alpha'_L$ the corresponding path opening on $\mathcal{P}(E')$, where $E' = E \cup B$ denotes the enlarged window. Thus we can compute $\alpha'_L$ according to the algorithm given in the previous sections. Define the opening $\alpha_L$ on $\mathcal{P}(E)$ by

$$\alpha_L(X) = \alpha'_L(X \cup B) \cap E.$$

In Fig. 6 we show that in this case, two paths which were originally disjoint may considered to be part of the same path which lies partially outside the window.

## 7. Incomplete Path Openings

### 7.1. *Motivation*

In practical applications, to increase the discriminatory power of path openings one might need to increase $L$. However as $L$ increases so does the probability of any path containing noise. To increase the robustness and flexibility of path openings, it is useful to allow a limited number of noise pixels to be ignored. This is the basic idea behind rank-max openings [4], which have been proven to work well in filtering applications [3], in particular with line structuring elements [8].

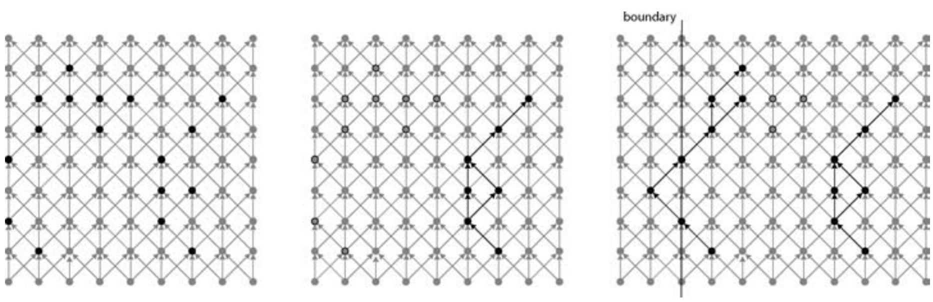



*Figure 6.* Opening $\alpha_6(X)$ of set $X$ (left) according to option $(a)$ (middle) and $(d)$ (right).

### 7.2. Framework

The path opening $\alpha_L(X)$ of a set $X$ comprises the union of all length-$L$ paths contained inside $X$. We can relax this condition by allowing up to $k$ vertices to be outside $X$. That is, we demand that at least $L-k$ out of $L$ vertices of the path must lie inside $X$. This yields a so-called *incomplete path opening* $\alpha_L^k(X)$. We present a formal definition below.

Define $\Pi_L^k(X)$ as the collection of length-$L$ paths in $E$ which contain at most $k$ points outside $X$:

$$\Pi_L^k(X) = \{a \in \Pi_L : |\sigma(a) \cap X^c| \le k\}.$$

Note that this definition only makes sense for $0 \le k \le L$, and that

$$\Pi_L(X) = \Pi_L^0(X) \subseteq \Pi_L^1(X) \subseteq \cdots \subseteq \Pi_L^{L-1}(X) \subseteq \Pi_L^L(X) = \Pi_L.$$

For $0 \le k \le L-1$ define the *incomplete path opening*

$$\alpha_L^k(X) = \bigcup \left\{\sigma(a) \cap X : a \in \Pi_L^k(X)\right\}.$$

It is obvious that

$$\alpha_L^0 \le \alpha_L^1 \le \cdots \le \alpha_L^{L-1},$$

that $\alpha_L^0 = \alpha_L$ and

$$\alpha_L^{L-1}(X) = \{x \in X : \Lambda(x) \ge L\},$$

where $\Lambda(x)$ was defined in (5). Putting

$$\bar{E}_L = \{x \in E : \Lambda(x) \ge L\},$$

we get that

$$\alpha_L^{L-1}(X) = X \cap \bar{E}_L.$$

Furthermore, for $0 \le k \le L$ we define

$$\psi_L^k(X) = \left\{a_1 : a \in \Pi_L^k(X)\right\}.$$

We have $\psi_L^0 = \psi_L$ and

$$\psi_L^L(X) = \{x \in E : \check{\lambda}(x) \ge L\} = \bigcup_{k \ge L} \check{E}_k.$$

Note that $\psi_L^k$ is defined for $0 \le k \le L$ while $\alpha_L^k$ is defined only for $0 \le k \le L-1$.

We will now express $\psi_{L+1}^{k+1}$ in terms of $\psi_L^{k+1}$ and $\psi_L^k$. Observe that $x \in \psi_{L+1}^{k+1}(X)$ if there exists $a = (a_1, \ldots, a_L)$ such that $(x, a_1, \ldots, a_L) \in \Pi_{L+1}$ and either $x \in X$ and $a \in \psi_L^{k+1}(X)$ or $a \in \psi_L^k(X)$. We have shown that

$$\psi_{L+1}^{k+1} = \left(\mathsf{id} \wedge \check{\delta}\psi_L^{k+1}\right) \vee \check{\delta}\psi_L^k. \tag{7}$$

This equation holds for $k = 0, 1, \ldots, L-1$. If we take $\psi_L^{-1}$ to be the empty set, then the equation holds also for $k = -1$, and in fact reduces to (4).

We now derive a more general form of (3). Consider a point $x \in \alpha_L^k(X)$, where $0 \le k \le L-1$. Thus $x \in X$ and there is a path $a \in \Pi_L$ with $x \in \sigma(a)$ such that $|\sigma(a) \cap X^c| \le k$. Assume that $a_l = x$. Now $a$ is the concatenation of the sequences $\boldsymbol{b} = (a_1, \ldots, a_{l-1}, x)$ and $\boldsymbol{c} = (x, a_{l+1}, \ldots, a_L)$. Define $j = |\sigma(\boldsymbol{b}) \cap X^c|$ which implies $|\sigma(\boldsymbol{c}) \cap X^c| \le k - j$. We conclude that

$$x \in \check{\psi}_l^j(X) \cap \psi_{L-l+1}^{k-j}(X).$$

Since the length of $\boldsymbol{b}$ is $l$ and $x \in X$ we have $0 \le j \le l-1$. Similarly since the length of $\boldsymbol{c}$ is $L+1-l$ we have $0 \le k-j \le L-l$. Together these imply that $0 \le j \le k$ and

$$j+1 \le l \le L+j-k$$

and we conclude that

$$\alpha_L^k = \mathsf{id} \wedge \bigvee_{j=0}^{k} \bigvee_{l=j+1}^{L+j-k} \left(\check{\psi}_l^j \wedge \psi_{L-l+1}^{k-j}\right) \tag{8}$$

for $0 \le k \le L-1$. Observe that this expression reduces to the one in (3) if $k = 0$.

The following is an algorithm for computation of the incomplete path opening. The computational cost for this algorithm is $O(kLN)$, with $N$ the number of elements in space $E$.

**Algorithm 3** (Incomplete Binary Path-Opening).

// Initialisation of bottom rows ($j = 0$) and diagonals ($j = l$)
// of arrays $Y_l^j = \psi_l^j(X)$ and $\check{Y}_l^j = \check{\psi}_l^j(X)$.
// Remainder of arrays $Y_l^j$ and $\check{Y}_l^j$.
// Using Eq. (7).
**for** $j = 0$ **to** $k-1$

**for** $l = j$ **to** $L + j - k - 1$
$Y_{l+1}^{j+1} = \left(X \cap \check{\delta} Y_l^{j+1}\right) \cup \check{\delta} Y_l^j$
$\check{Y}_{l+1}^{j+1} = \left(X \cap \delta \check{Y}_l^{j+1}\right) \cup \delta \check{Y}_l^j$
**end**
**end**
// Construction of incomplete opening $A_L^k = \alpha_L^k(X)$
// Using Eq. (8).
$A_L^k = \emptyset$
**for** $j = 0$ **to** $k$
**for** $l = j + 1$ **to** $L + j - k$
$A_L^k = A_L^k \cup \left(Y_l^j \cap \check{Y}_{L-l+1}^{k-j}\right)$
**end**
**end**
$A_L^k = A_L^k \cap X$

### 7.3. *Illustration*

Figure 7 is a real example of the use of incomplete path openings in the binary case. In this image we have a thin glass fibre observed under an electron microscope with some noise present. A normal (i.e. complete) path opening deletes most of the noise together with the middle part of the fibre. An incomplete path opening with a small tolerance still deletes most of the noise while keeping the middle part of the fibre untouched.

The result of an incomplete path closing in the grey-level case is shown on Fig. 8(e) and discussed in the final section.

## 8. The Grey-Scale Case

In this section we extend the results developed in the previous sections to the grey-scale case, with the exception of the opening transform which is not defined in the grey-scale case to the best of our knowledge.

First we define the grey-scale analogue, $\Pi_L^t(I)$, of the path collection $\Pi_L(X)$. This involves an additional parameter $t$ representing the grey-level of the path:

$$\Pi_L^t(I) = \{a \in \Pi_L : I(a_k) \geq t,\ k = 1, 2, \ldots, L\}$$

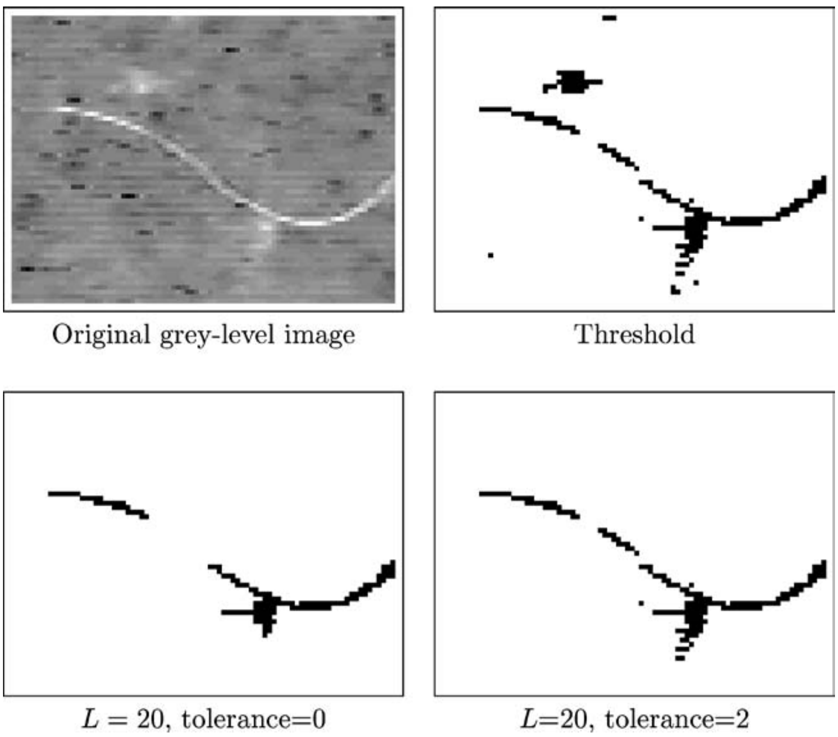
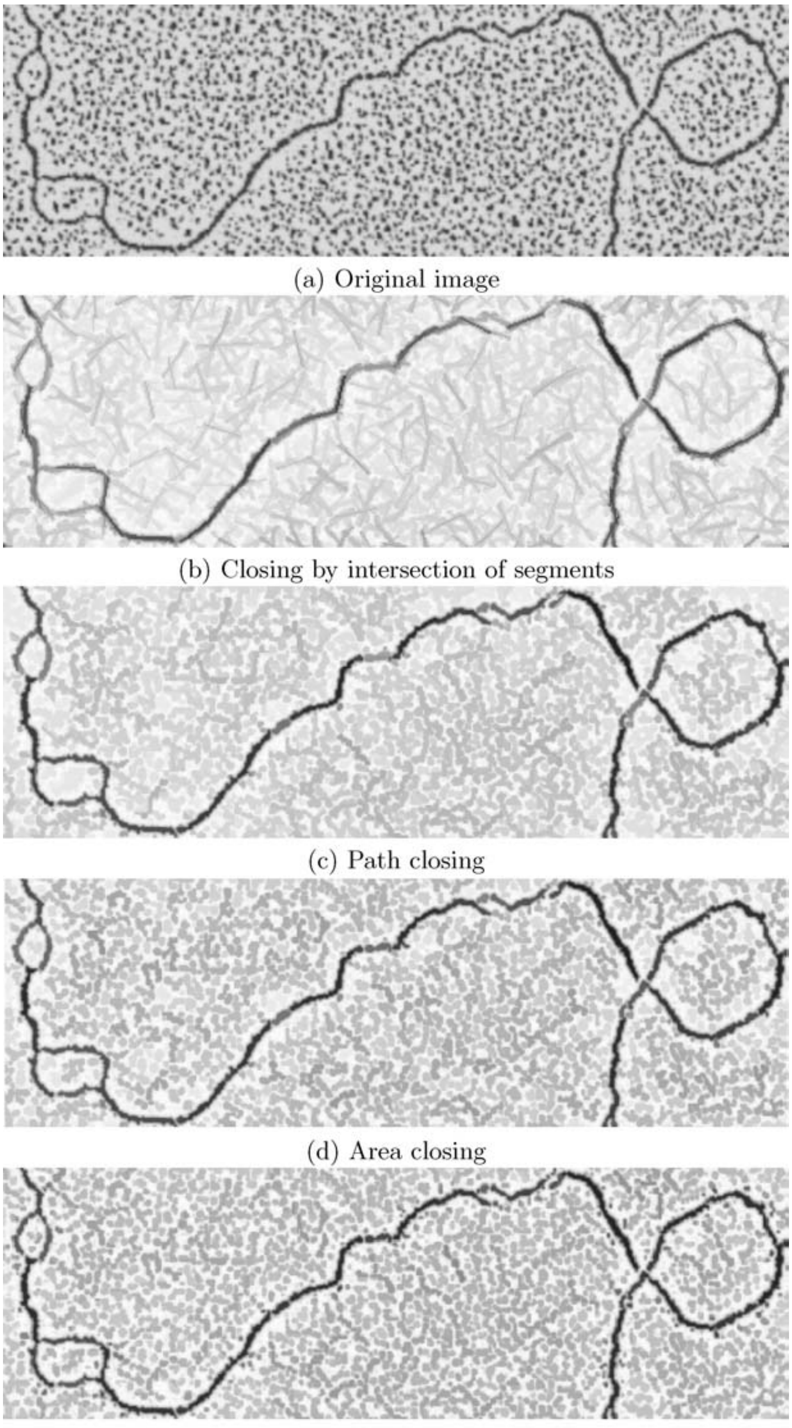



*Figure 7.* Glass fibre electron micrograph (a) and a threshold (b). Comparison between complete (c) and incomplete (d) path opening. With the incomplete path opening more of the fibre is retained.



(a) Original image

(b) Closing by intersection of segments

(c) Path closing

(d) Area closing

(e) Incomplete path closing

*Figure 8.* Example of path closing compared with closing with segments and area closing.

where

It is easy to see that

$$\Pi_L^t(I) = \Pi_L(X_t(I))$$

where

$$X_t(I) = \{x \in E : I(X) \geq t\}$$

is the *level set* of $I$ at value $t$. Let $\Psi_k$ be the flat extension of $\psi_k$ defined by means of the level set construction, and let $\mathcal{A}_L$ be the flat extension of $\alpha_L$.

For a domain $E$ and range $T$ define $\mathrm{Fun}(E, T) = T^E$ to be the space of grey-level functions from $E$ to $T$.

**Proposition 8.1.** *The operator $\Psi_k$ and the opening $\mathcal{A}_L$ on Fun$(E, T)$ are, respectively, given by*

$$\Psi_k(I)(x) = \max\{t \in T : (x, a_2, \ldots, a_k) \in \Pi_k^t(I) \text{ for some } a_2, \ldots, a_k \in E\}$$

$$\mathcal{A}_L(I)(x) = \max\{t \in T : x \in \sigma(\boldsymbol{a}) \text{ for some } \boldsymbol{a} \in \Pi_L^t(I)\}$$

From the theory on flat function operators [2] we know that the expressions in (3) and (4) carry over immediately to the function case. Therefore the grey-scale opening $\mathcal{A}_L(I)$ of an image $I$ may be computed by an algorithm which is a straightforward generalisation of Algorithm 1. Specifically, in Algorithm 1 we replace $X$ by $I$, $\emptyset$ by $-\infty$, $\cup$ by $\vee$, $\cap$ by $\wedge$ and the dilations $\delta$ and $\breve{\delta}$ by their flat extensions.

## 9. Example

Figure 8 is an example of path closing compared with other methods. We chose to illustrate with a closing rather than an opening because of the better contrast in the printing process, but the same conclusions would apply to both. The parameter choices for each operation cannot be made strictly identical, we chose them so that the outputs would be comparable.

Figure 8(a) is the grey-level original $500 \times 160$ image. This is an image of DNA (the long thin structure) observed in a scanning electron microscope. The objective is to separate the DNA from the noisy background, and we use various closings as pre-processing filters.

Figure 8(b) is the result of applying a closing by intersection of 44 segments of length 23 pixels, each in a different direction, approximately uniformly oriented (subject to the digital grid). As can be seen, after the application of this closing the background is mostly suppressed, but so is some of the DNA.

Figure 8(c) is a path closing with path length of 33 pixels, the specific form of which is given below. While this is longer than that of the straight segments family in (b), the shape and contrast of the DNA is well preserved, while the background is substantially suppressed.

Figure 8(d) is an area closing with parameter 50 square pixels. The DNA is well preserved as in (c) but the background is not as effectively filtered out.

Figure 8(e) is an incomplete path-closing with path length 33 and tolerance 2. This allows more of the DNA to be preserved, however the result is that some of the noise close to the DNA is not filtered out.

As can be seen, the path closing (c) was able to better preserve the shape of the object of interest than the closing with segments, while removing more of the unwanted background than the area closing, which was the intended behaviour.

To produce the closing shown in Fig. 8(c), four closings were computed using the grey-scale form of Algorithm 1. The pixelwise minimum of these was then obtained, giving a final result which is still therefore a closing. In the first two instances, the adjacency graphs were those of Fig. 2(a) and (c)—that is, adjacencies directed towards the north and north-east respectively. The other two closings were those based on similar adjacencies directed towards the east and south-east.

The result of this combination was to choose a family of paths such that at each point the entirety of the path was contained in a 90 degree angle double-ended cone, either vertically or diagonally. In some sense this captures the idea of a family of oriented, but flexible structuring elements. It is of course possible to modify the adjacency graphs in order to constrain these paths more or less.

## 10. Conclusions

In this paper we have explored the theory of path openings and closings on binary and grey-level images. Path openings are openings over a large number of connected or disconnected paths, which extend the useful notions of openings by unions of line segments by allowing the use of oriented, narrow but non-straight segments as a family of structuring elements. Because of the oriented nature of the family of structuring elements used, the resulting operators are more constrained than area openings.

Path openings and closings essentially allow practitioners to close the gap between openings by line segments (which are constrained and anisotropic) and area openings (which are unconstrained and isotropic). The framework developed in this paper allows for paths which behave more closely like one or the other, by varying the adjacency relation.

We have developed a workable solution for such path openings with low complexity, which makes the computation of such paths practical.

Finally we have explored the questions of how to deal with border effects, how to compute path opening transforms (only in the binary case) and how to extend path openings to incomplete paths, which could provide a degree of robustness against noise.

## Note

1. 'Reflexive' would mean that $x \mapsto x$ for every $x \in E$. 'Symmetric' would mean that $x \mapsto y$ iff $y \mapsto x$, for every $x, y \in E$.

## References

1. M. Buckley and H. Talbot, "Flexible linear openings and closings," in *Mathematical Morphology and its Application to Image Analysis*, L. Vincent and D. Bloomberg (Eds.), Palo Alto, ISMM, Kluwer, June 2000.
2. H.J.A.M. Heijmans, *Morphological Image Operators*, Academic Press: Boston, 1994.
3. H.J.A.M. Heijmans, "Morphological filters for dummies," in *Mathematical Morphology and its Applications to Image and Signal Processing*, Petros Maragos, Ronald W. Schafer, and Muhammad A. Butt (Eds.), Atlanta, GA, Proceedings for ISMM'96, Kluwer Acad., May 1996, pp. 127–137.
4. C. Ronse, "Extraction of narrow peaks and ridges in images by combination of local low rank and max filters: Implementation and applications to clinical angiography," Working Document WD47, Philips Research Laboratories, Brussels, Belgium, 1988.
5. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press: London, 1982.
6. J. Serra (Ed.), *Image Analysis and Mathematical Morphology. II: Theoretical Advances*, Academic Press: London, 1988.
7. P. Soille, *Morphological Image Analysis, Principles and Applications*, Springer, 1999.
8. P. Soille and H. Talbot, "Directional morphological filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 11, pp. 1313–1329, 2001.
9. M. Vandroogenboeck and H. Talbot, "Fast computation of morphological operations with arbitrary structuring elements," *Pattern Recognition Letters*, Vol. 17, pp. 1451–1460, 1996.
10. L. Vincent, "Minimal path algorithms for the robust detection of linear features in images," in *Mathematical Morphology and its Applications to Image and Signal processing*, vol. 12 of *Computational Imaging and Vision*, Amsterdam, Kluwer, Proceedings for ISMM'98, June 1998, pp. 331–338.

**Henk Heijmans** was born in 1957 in Tilburg, The Netherlands. After obtaining his masters degree in mathematics from the Technical University in Eindhoven, he went to the Centre for Mathematics and Computer Science where he became a PhD student working in the area of mathematical biology. He received his PhD degree from the University of Amsterdam in 1985. Currently Henk Heijmans is heading the "Signals and Images" research group (PNA4) at CWI, Amsterdam, The Netherlands. The aim of this research theme is to combine expertises in various directions in signal and image analysis, in which mathematics and computer science play an essential role, thereby contributing to solutions of important problems for society in the coming years. His own research interests are focused towards mathematical techniques for image and signal processing, with an emphasis on mathematical morphology and wavelet analysis.

**Michael Buckley** is a Principal Research Scientist at CSIRO, Mathematical and Information Sciences, with the Biotechnology and Health Informatics research program. He has a Ph.D. in Mathematical Statistics. In recent years Michael has published several papers on microarray normalisation. In 2003–2004 he has worked in proteomics, including protein chips, LC-MS-MS and biomarker discovery. He has substantial experience in dynamic programming, Hidden Markov Models and related methods for matching and modelling amino acid or nucleotide sequences. He has also worked in the field of image analysis, including the Road Crack project where he was the technical leader of algorithm development, collaborations with Axon Instruments Inc. and analysis of cDNA microarray images.

**Hugues Talbot** received the engineering degree from École Centrale de Paris in 1989, the D.E.A. (Masters) from University Paris VI in 1990 and the Ph.D from École des Mines de Paris in 1993, under the guidance of Dominique Jeulin and Jean Serra. He has been a scientist with the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Mathematical and Information Sciences since 1994. He has worked on numerous applied projects in relation with industry, has contributed more than 30 research papers in international journals and conferences and has co-edited two sets of international conference proceedings on image analysis. He was the general chair for DICTA'2003, the largest regular image processing conference in Australia. He now also teaches image processing at the University of Sydney, and his research interests include image segmentation, linear structure analysis, texture analysis and algorithms.