EFFICIENT PATH OPENINGS AND CLOSINGS

Ben Appleton[1] and Hugues Talbot,[2]

[1] *School of Information Technology and Electrical Engineering,*
The University of Queensland
Brisbane, Australia
appleton@itee.uq.edu.au

[2] *CSIRO Mathematical and Information Sciences*
Locked Bag 17, North Ryde NSW 2113 Australia
Hugues.Talbot@csiro.au

For Henk.

Abstract Path openings and closings are algebraic morphological operators using families of thin and oriented structuring elements that are not necessarily perfectly straight. These operators are naturally translation invariant and can be used in filtering applications instead of operators based on the more standard families of straight line structuring elements. They give similar results to area or attribute-based operators but with more flexibility in the constraints.

Trivial implementations of this idea using actual suprema or infima of morphological operators with paths as structuring elements would imply exponential complexity. Fortunately a linear complexity algorithm exists in the literature, which has similar running times as an efficient implementation of algebraic operators using straight lines as structuring elements.

However even this implementation is sometimes not fast enough, leading practitioners to favour some attribute-based operators instead, which in some applications is not optimal.

In this paper we propose an implementation of path-based morphological operators which has logarithmic complexity and comparable computing times with those of attribute-based operators.

Keywords: Algebraic morphological operators, attributes, complexity.

Introduction

One key concept of morphological filtering is that due to the non-linearity of morphological operators, information is lost irretrievably during the filtering process. Morphological filter design often consists of using the right composition of operators to remove the noise and unwanted features and to preserve the desired information as much as possible.

Many problems in image analysis involve oriented, thin, line-like objects, for example measuring thin fibres [22, 20], hair detection [19, 15], blood vessel detection [8], grid-line extraction on stamped metal pieces [21] and others.

Definitions

Within the framework of Mathematical Morphology, given a complete lattice $\mathcal{L}$ with ordering relation $\succeq$, an opening γ is a transform $\mathcal{L} \longrightarrow \mathcal{L}$ which is increasing, idempotent and anti-extensive. Similarly a closing ϕ is a transform which is increasing, idempotent and extensive [4].

Following the definitions in [16], a dilation δ_B using structuring element B of element f in $\mathcal{L}$ is $\delta_B(f) = \bigvee_{b \in B} f_{-b}$ and an erosion is defined similarly by $\epsilon_B(f) = \bigwedge_{b \in B} f_{-b}$.

Morphological opening $\gamma_B(f)$ is defined by $\gamma_B(f) = \delta_{\check{B}}[\epsilon_B(f)]$ and morphological closing $\phi_B(f)$ by $\phi_B(f) = \epsilon_{\check{B}}[\delta_B(f)]$. It is easy to confirm that γ_B and ϕ_B have all the properties of openings and closings respectively.

Algebraic openings and closings are those transforms which have the three properties mentioned above but which are not necessarily defined as the simple composition of one dilation and one erosion by a given structuring element.

Suprema of openings are openings and infima of closings are closings [14]. Composing openings and closing in these fashions are common ways of producing algebraic openings and closings.

Morphological operators have many interesting properties that are well described for example in [7, 13]. In particular openings and closings are dual operators, and so in the following we will only discuss the opening without loss of generality.

Morphological operators for thin structures

In an application where some bright, thin and elongated structure needs to be segmented, one typical approach is to remove the features in the image which are neither thin nor elongated. If the structures are also bright on a dark background, the standard approach would be to use an infimum of openings using lines as structuring elements oriented in many directions [10]. The result is an isotropic transform if the line structuring element lengths are adjusted to be independent of orientation [11].

The implementation of such an operator with actual lines as structuring elements is inefficient, however using recursive implementations of openings at arbitrary angles yields a linear time algorithm [17] with respect to the length of the structuring elements. Note that this algorithm is not translation-invariant. A translation-invariant version, which should be used if features are very thin, was proposed in [18] which is more expensive but still of linear complexity.

Area and attributes openings [1, 12, 23] are also often used for the analysis of thin structures. An area opening of parameter λ is equivalent to the supremum of all the openings by connected structuring elements of area λ. Obviously this includes all the straight line structuring elements of this length.

Practitioners often note that using only straight line structuring elements removes too much of the desired features, while using area operators does not allow them to distinguish between long and narrow features on the one hand, and short compact ones on the other. While it is sometimes possible to combine these operators to obtain the desired effect of retaining thin narrow structures while filtering out compact noise, this is not always possible.

Recently efficient morphological operators equivalent to using families of narrow, elongated but not necessarily perfectly straight structuring elements were proposed in [3] and [6], together with an algorithm for computing the transform of linear complexity with regards to the length of the structuring elements. These path operators constitute a useful medium between operators using only straight lines and those using area or other attributes.

In the remainder we propose a significantly faster algorithm for implementing path operators, with logarithmic complexity with respect to the length of the structuring elements.

1. Path-based morphological opening

The theory of path openings is explained in detail in [6] and in a shorter fashion in [5]. We only summarize the main points here.

1.1 Adjacency and paths

Let E be a discrete 2-D image domain, a subset of $\mathbb{Z}^2$. Then $\mathcal{B} = \mathcal{P}(E) = 2^E$ is the set of binary images and $\mathcal{G} = \mathcal{T}^E$ the space of grey-scale functions, where $\mathcal{T}$ is the set of grey values. We assume E is endowed with an adjacency relation $x \mapsto y$ meaning that there is a directed edge going from x to y. Using the adjacency relation we can define the dilation $\delta(\{x\}) = \{y \in E, x \mapsto y\}$. The L-tuple $\boldsymbol{a} = (a_1, a_2, ..., a_L)$ is called a *δ-path of length L* if $a_{k+1} \in \delta(\{a_k\})$ for $k = 1, 2, ..., L-1$. Given a path $\boldsymbol{a}$ in E, we denote by $\sigma(\boldsymbol{a})$ the set of its elements, i.e: $\sigma(a_1, a_2, ..., a_L) = \{a_1, a_2, ..., a_L\}$. We denote the set of all δ-paths of length L by Π_L, and the set of δ-paths of length L contained in a subset X of E is denoted by $\Pi_L(X)$.

1.2 Path openings

We define the operator $\alpha_L(X)$ as the union of all δ-paths of length L contained in X:

$$\alpha_L(X) = \bigcup\{\sigma(\boldsymbol{a}), \boldsymbol{a} \in \Pi_L(X)\} \tag{1}$$

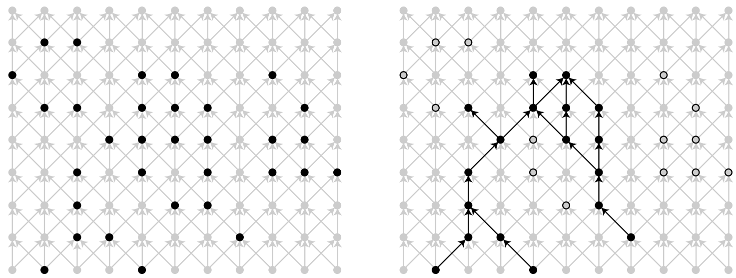

Figure 1. A set $X \subseteq E$ (black points on the left) and its opening $\alpha_6(X)$ (black points on the right). Unfilled points on the right have been discarded. The underlying adjacency graph is in light grey.

It is easy to establish that α_L has all the properties of an opening. Figure 1 offers an illustration. For an adjacency graph similar to that of this figure, and for an unbounded image, there are 3^{L-1} distinct paths of length L starting from any point. The path opening α_L is in fact the supremum of the morphological opening using these paths as structuring elements, which would suggest an inefficient way to compute the transform. Fortunately [6] proposes a useful recursive decomposition which allows the transform α_L to be computed in linear time with respect to L (not presented here due to lack of space).

1.3 Grey-level transform and practical considerations

The binary transform defined above extends to the grey-level domain in the usual way by replacing the union with a supremum. The recursive decomposition in [6] also extends to the grey-level domain.

The current definition of a path opening with an adjacency graph such as in Fig. 1 is not sufficiently useful in a context where features are distributed isotropically : only paths generally oriented North – South will be preserved by the opening. We need to take a supremum with openings using adjacency graphs oriented East – West, North-East – South-West and North-West – South-East. More complex adjacency graphs can also be devised for more constrained path operators, however in the remainder we assume this basic scheme.

2. Ordered algorithm

The grayscale path opening algorithm presented in this paper is based on a few simple observations. Firstly, the principle of threshold decomposition allows the construction of grayscale morphological operators from binary morphological operators. Secondly, in the case of grayscale path openings it is possible to efficiently compute the set of binary path openings for all thresholds in sequence.

2.1 Threshold decomposition

Here we equivalently redefine binary images as functions of the form $b : E \to \{\mathbf{false}, \mathtt{true}\}$ rather than subsets of the image domain E. Then, given a grayscale image $g \in G$, a threshold operator $T_t : \mathcal{G} \to \mathcal{B}$ with threshold t, and a binary opening $\gamma_B : \mathcal{B} \to \mathcal{B}$, there exists a grayscale opening $\gamma_G : \mathcal{G} \to \mathcal{G}$ such that for all thresholds t we have $T_t \circ \gamma_G = \gamma_B \circ T_t$, where $\circ$ is the composition operator.

This grayscale opening $\gamma_G(g)$ may be constructed explicitly by 'stacking' the results of the binary opening applied to each threshold of the original image. This stacking assigns to each pixel p the highest threshold t for which the binary opening $\gamma_B \circ T_t(g)$ remains $\mathtt{true}$.

2.2 Updating binary path openings

The second observation is that the binary images produced in this construction tend to vary little between sequential thresholds. In the case of path openings we will show how to efficiently update the result of the binary opening $\gamma_B \circ T_t(g)$ from the result of the binary opening at the previous threshold $\gamma_B \circ T_{t-1}(g)$.

For brevity we here describe only the case of North – South paths. In this case the path opening transform of a binary image b stores at each pixel p two values: the length $\lambda^-[p]$ (not including p itself) of the longest path travelling upward from pixel p, and the length $\lambda^+[p]$ of the longest path travelling downward from pixel p. Then the length of the longest path passing through pixel p (where $b[p] = \mathtt{true}$) is $\lambda[p] = \lambda^-[p] + \lambda^+[p] + 1$. If $b[p] = \mathbf{false}$ then we define $\lambda[p] = 0$. An algorithm for computing the opening transform via the recursive computation of λ^- and λ^+ has previously been described in [6]. In short, we may state that in the North – South case

$$\lambda^-[p] = 1 + \max(\lambda^-[(p^1 - 1, p^2 - 1)],\ \lambda^-[(p^1, p^2 - 1)],\ \lambda^-[(p^1 + 1, p^2 - 1)]) \quad (2)$$

and

$$\lambda^+[p] = 1 + \max(\lambda^+[(p^1 - 1, p^2 - 1)],\ \lambda^+[(p^1, p^2 - 1)],\ \lambda^+[(p^1 + 1, p^2 - 1)]) \quad (3)$$

where $b[p] = \mathtt{true}$, and 0 otherwise.

Now, in order to update the binary opening $\gamma_B \circ T_t(g)$ given the result from the previous threshold $\gamma_B \circ T_{t-1}(g)$, we must compute the new binary opening transform λ and hence λ^- and λ^+. Rather than recomputing these from the image $b = T_t(g)$, we may compute the changes to λ^- and λ^+ due solely to the pixels which made the transition from $\mathtt{true}$ to $\mathbf{false}$ between $T_{t-1}(g)$ and $T_t(g)$. This is performed in the following steps:

- Initialisation:
  - Set all pixels with $g[p] = t$ to *active* and enqueue
- For each row from top to bottom:
  - For all *active* pixels p in this row:
    - Recompute $\lambda^-[p]$ according to Equation 2
    - If $\lambda^-[p]$ changed, set as *active* and enqueue the dependent pixels $(p^1-1, p^2+1), (p^1, p^2+1), (p^1+1, p^2+1)$
- For each row from bottom to top:
  - For all *active* pixels p in this row:
    - Recompute $\lambda^+[p]$ according to Equation 3
    - If $\lambda^+[p]$ changed, set as *active* and enqueue the dependent pixels $(p^1-1, p^2-1), (p^1, p^2-1), (p^1+1, p^2-1)$

The queueing system for *active* pixels consists of a first-in-first-out (FIFO) queue for each row as well as a queue of rows which contain active pixels. This queueing system is necessary to comply with the dependencies in Equations 2 and 3 and also avoids inefficiently scanning the entire image.

2.3 Recursive ordered path opening

Here we present an algorithm to compute a grayscale path opening. L denotes the desired path length. Note that, as we are interested in the specific path length L, path lengths λ^-, λ^+ greater than $L-1$ may be treated as equal to $L-1$ in Algorithm 2.2. This limits the propagation of changes to the binary opening transform and hence improves the efficiency of the grayscale path opening.

- Initialisation:
  - Sort the pixels by their intensities
  - Set $b[p] = \mathbf{true}$ for all pixels p
  - Compute λ^+, λ^- from b.
- For each threshold t in $\mathcal{T}$:
  - Using Algorithm 2.2, update λ^-, λ^+ for the new threshold
  - For all *active* pixels p whose path length $\lambda[p]$ consequently dropped below L, set $\gamma_G(g)[p] = t$

Sorting the pixels by their intensities is a necessary preprocessing step in order to efficiently locate the pixels whose threshold changes in the step $t-1 \rightarrow$

t. For integer data a linear-time sorting algorithm such as the Radix sort is recommended [9]. Alternatively a suitable priority queue data structure [2] can be used.

A simple heuristic has been found to further improve the efficiency of this algorithm in practice. When the maximal path length of a pixel p drops below L, it cannot contribute to a path of length L or greater at any further threshold. Therefore we may remove this pixel from further consideration by setting $b[p] = \textbf{false}$. We refer to this as the *length heuristic* in the remainder of this paper. We believe that the average running time of this algorithm is $O(N \log L)$ on images containing N pixels. However the formal derivation of this average running time would require the selection of an appropriate stochastic image model and is not pursued in this paper.

3. Opening transform

The algorithm presented in Section 2.3 may be extended in a simple manner to compute the grayscale path opening transform. To the authors' knowledge this is the first presentation of an opening transform for grayscale images.

In the course of Algorithm 2.3, the path opening transforms for all binary thresholds were computed in sequence. Instead of discarding these intermediate results we may store them in compressed form allowing them to be queried at a later point. At each threshold, those *active* pixels whose maximal path length $\lambda[p]$ has decreased store a point $(t, \lambda[p])$ in a linked list. This linked list is monotonically increasing in t and monotonically decreasing in $\lambda[p]$. Once computed, we may query this structure with any desired path length to extract the associated grayscale path opening.

- Initialisation: As per Algorithm 2.3
- For each threshold t in $\mathcal{T}$:
  - Using Algorithm 2.2, update λ^-, λ^+ for the new threshold.
  - For all *active* pixels p whose path length $\lambda[p]$ decreased, append the node $(t, \lambda[p])$ to the linked list at pixel p.

This algorithm requires the same order of computation as Algorithm 2.3, that is $O(N \log L)$. The number of linked list nodes generated in Algorithm 3 must be less than the number of operations in Algorithm 2.3, and therefore the average memory required by Algorithm 3 is $O(N \log L)$. This may also suggest that path openings are inherently informative, as we may store all path openings in $O(\log L)$ bytes per pixel rather than $O(L)$ as may be initially expected.

4. Results

op
an
be
th
lin
pa

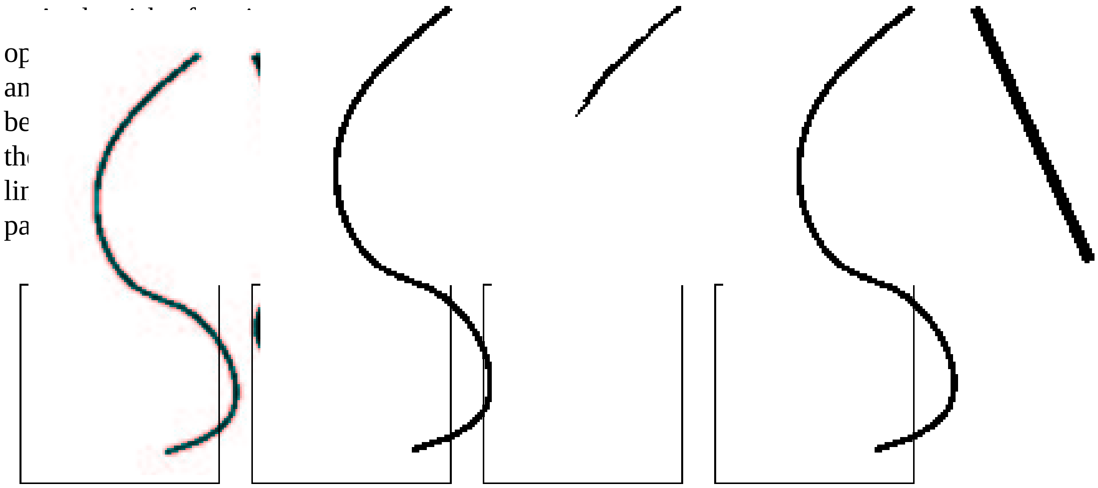

Figure 2. Toy example: On the input we wish to retain the line-like features while eliminating compact noise. Only the path opening works in this case.

A more useful example is shown in Fig. 3. We wish to detect the small thin fibres in this electron micrograph present at the bottom of this image. The large fibres are detected by a different method [20] which is of no interest here. The thin fibres are present on a noisy background which requires some filtering. A supremum of openings by lines is too crude here (result not shown). An area opening does not eliminate enough of the noise, but a path opening works as expected.

4.1 Timings

Table 1 shows the running times of the proposed algorithm compared with various alternatives. We observe that the proposed ordered path opening implementation has a running time approximately logarithmic (plus a constant) with respect to L, while both the recursive path opening and the supremum of openings by lines have approximately linear running times. Note that the individual openings by lines in the latter algorithm are all running in constant time irrespective of L, but for larger L more orientations need to be explored. Note also that the presented algorithm for the supremum of opening by lines is a non-translation-invariant implementation. A translation-invariant version would be significantly slower still. The area opening algorithm seems to converge to a constant-time algorithm with low constant. The area parameter was simply L, although a constant times L could have been chosen, e.g. $3 \times L$ without significantly affecting the result.

Co

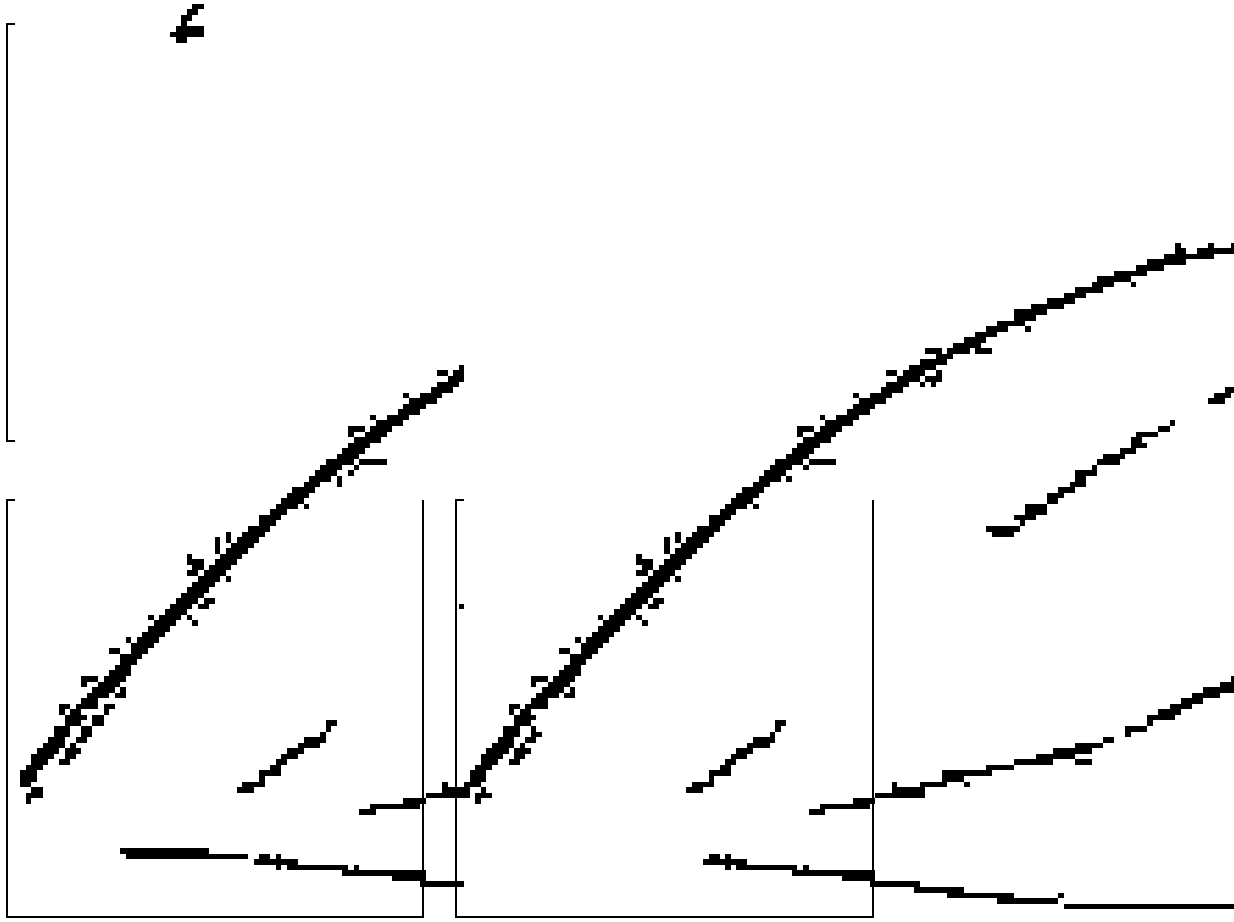

Figure 3. Electron micrograph of glass fibres: to detect the small thin fibres in the bottom of the image, a white top-hat is useful but noisy. When filtered by an area opening some compact noise remain while a path opening yields a better result.

Memory demands for these algorithms are all low except the recursive path opening implementation which requires an amount of memory proportional to LN, with N the number of pixels in the image.

We observe that the area opening is the fastest algorithm by far, but that the presented path opening algorithm comes second, a factor of between 4 and 5 times slower than the area opening, but significantly faster than the other two algorithms for most useful values of L.

Table 1. Comparison of algorithm running times. From left to right the columns are the ordererd path opening presented in this paper, the recursive path opening of Heijmans et. al, the supremum of openings by lines and the area opening. Timings are in seconds, image was 560×510×8-bit. Processor was a Pentium IV 1.7GHz.

L	Ordered PO	Recursive PO	Supremum Lines	Area
1	0.56	0.08	0.14	0.13
5	0.69	0.54	0.65	0.17
10	0.73	1.16	1.38	0.17
50	0.90	14.24	3.29	0.21
100	0.93	30.74	11.43	0.22

5. Conclusion and future work

We have presented a new, ordered implementation of the path opening and closing transforms. This transform is identical to the supremum (resp. infimum) of openings by a family of structuring elements described as oriented paths. The family of paths is of exponential size with respect to their length L, but there exists a recursive implementation with only linear complexity with respect to L.

We have proposed a new implementation with only logarithmic complexity with respect to L, which is observed to be much faster than the recursive implementation except for very small L. It is also faster than the usual transform by unions of lines structuring elements used for the study of thin structures.

The area transforms are still faster than the proposed implementation, by a nearly constant factor of 4 to 5. However, the proposed algorithm is fast enough for many applications and can be used in cases where using an area or attribute transform is not appropriate, e.g. in the presence of sufficiently large compact noise.

The path transforms are intuitive, translation-invariant methods useful for the analysis of thin, elongated but not necessarily perfectly straight structures.

Future work will include incomplete path openings, i.e. paths which are not necessarily connected.

References

[1] E. Breen and R. Jones. Attribute openings, thinnings and granulometries. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 64(3):377–389, 1996.

[2] E. J. Breen and D. Monro. An evaluation of priority queues for mathematical morphology. In J. Serra and P. Soille, editors, *Mathematical morphology and its applications to image processing*, pages 249–256. Kluwer, 1994.

[3] M. Buckley and H. Talbot. Flexible linear openings and closings. In L. Vincent and D. Bloomberg, editors, *Mathematical Morphology and its application to image analysis*, Palo Alto, June 2000. ISMM, Kluwer.

[4] H. Heijmans. *Morphological image operators*. Advances in Electronics and Electron Physics Series. Academic Press, Boston, 1994.

[5] H. Heijmans, M. Buckley, and H. Talbot. Path-based morphological openings. In *ICIP'04*, Singapore, October 2004.

[6] H. Heijmans, M. Buckley, and H. Talbot. Path openings and closings. *Journal of Mathematical Imaging and Vision*, 2004. In press.

[7] H.J.A.M. Heijmans and C. Ronse. The algebraic basis of mathematical morphology, part I: dilations and erosions. *Computer Vision, Graphics, and Image Processing*, 50:245–295, 1990.

[8] F. Jaumier. Application de la morphologie mathématique à la détection de formes sur les clichés angiofluorographiques de la rétine. Master's thesis, July 1988.

[9] D. Knuth. *The art of computer programming. Volume 3: Sorting and searching*. Addison-Wesley, 1973.

[10] M.B. Kurdy and D. Jeulin. Directional mathematical morphology operations. In *Proceedings of the 5th European Congress for Stereology*, volume 8/2, pages 473–480, Freiburg im Breisgau, Germany, 1989. Acta Stereologica.

[11] C. L. Luengo Hendriks and van Vliet L. J. A rotation-invariant morphology for shape analysis of anisotropic objects and structures. In C. Arcelli, L.P. Cordella, and G. Sanniti di Baja, editors, *Visual Form 2001: 4th International Workshop on Visual Form*, page 378, Capri, Italy, 2001. Springer LNCS.

[12] A. Meijster and H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):484–494, April 2002.

[13] C. Ronse and H.J.A.M. Heijmans. The algebraic basis of mathematical morphology, part ii: openings and closings. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 54:74–97, 1991.

[14] J. Serra. *Image analysis and mathematical morphology*. Academic Press, 1982.

[15] V.N. Skladnev, A. Gutenev, S. Menzies, L.M. Bischof, H.G.F Talbot, E.J. Breen, and M.J. Buckley. Diagnostic feature extraction in dermatological examination. Technical Report International Publication Number WO 02/094098 A1, Polartechnics Limited, 2001.

[16] P. Soille. *Morphological Image Analysis, principles and applications*. Springer-Verlag, 2nd edition, 2003. ISBN 3-540-42988-3.

[17] P. Soille, E. Breen, and R. Jones. Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):562–567, May 1996.

[18] P. Soille and H. Talbot. Directional morphological filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1313–1329, 2001.

[19] Lee T., Ng V., Gallagher R., Coldman A., and McLean D. Dullrazor: A software approach to hair removal from images. *Computers in Biology and Medicine*, 27:533–543, 1997.

[20] H. Talbot, T. Lee, D. Jeulin, D. Hanton, and L. W. Hobbs. Image analysis of insulation mineral fibres. *Journal of Microscopy*, 200(3):251–268, Dec 2000.

[21] A. Tuzikov, P. Soille, D. Jeulin, H. Bruneel, and M. Vermeulen. Extraction of grid lines on stamped metal pieces using mathematical morphology. In *Proc. 11th IAPR International Conference on Pattern Recogni tion, Conference A: Computer Vision and Applications*, volume 1, pages 425–428, The Hague, September 1992.

[22] G. van Antwerpen, P.W. Verbeek, and F.C.A Groen. Automatic counting of asbestos fibres. In I.T. Young et al., editor, *Signal Processing III: theories and applications*, pages 891–896, 1986.

[23] L. Vincent. Grayscale area openings and closings, their efficient implementation and applications. In *Proceedings of the conference on mathematical morphology and its applications to signal processing*, pages 22–27, Barcelona, Spain, May 1993.