Algorithms for the Decomposition of Gray-Scale Morphological Operations

Ronald Jones and Imants Svalbe, *Member, IEEE*

***Abstract*— The choice and detailed design of structuring elements plays a pivotal role in the morphologic processing of images. A broad class of morphological operations can be expressed as an equivalent supremum of erosions by a minimal set of basis filters. Diverse morphological operations can then be expressed *in a single, comparable framework*. The set of basis filters are data-like structures, each filter representing one type of local change possible under that operation. The data-level description of the basis set is a natural starting point for the design of morphological filters.**

This paper promotes the use of the basis decomposition of gray-scale morphological operations to design and apply morphological filters. A constructive proof is given for the basis decomposition of general gray-scale morphological operations, as are practical algorithms to find all of the basis set members for these operations. Examples are given to illustrate the algorithms presented.

***Index Terms*—Gray-scale morphology, mathematical morphology, morphologic basis decomposition.**

NOMENCLATURE

f, g, k	the functions f, g, k
F, G, K	the sets F, G, K
x, y, z	n-tuple elements of a set
E^N	Euclidean N-space
$f : F \rightarrow E$	f maps F to E
$\theta(f), \phi(f), \psi(f)$	operations on an image f
$\ominus$	erosion
$\oplus$	dilation
$\circ$	open
$\bullet$	close
sup	the supremum operator
inf	the infimum operator

I. INTRODUCTION

GRAY-SCALE morphology uses roving surface patches, known as structuring elements, to explore, through local pattern matching, the shapes contained in gray-scale images. The structuring elements act as template-like shapes used to extract shape dependent information from signals or images. Overviews of gray-scale morphology can be found in [1]–[4], [20]; we use the definitions of the fundamental morphological operations given in [1].

Manuscript received October 7, 1991; revised June 24, 1993. Recommended for acceptance by Associate Editor J. J. Little.

The authors are with the Physics Department, Faculty of Science, Monash University, Clayton 3168, Australia; e-mail: imantsz@physics1.physics.monash.edu.au.

IEEE Log Number 9400020.

A. Previous Work

The foundation for this work is provided by Matheron's theorem [5] which demonstrates that certain broad classes of morphologic operations have an exact, parallel representation as a union of erosions (or dually as an intersection of dilations). Maragos [6] and Giardina and Dougherty [7] extended this theorem to gray-scale morphology, and introduced the concept of the basis as an equivalent and *minimal* description of an operation. Further extensions have been made by Banon and Barrera to all translation-invariant set mappings (not necessarily increasing) [17] and to all mappings between lattices [18]. The basis representation has seen practical implementation in the work of Dougherty and Loce [16], where it has been used to design optimal morphological filters, and in the work of Khosravi and Schafer [19], where it has been used to represent linear filters. Recent work [9] reports an algorithm for the minimal basis decomposition of the binary close operation and surveyed some of the properties of discrete basis sets and the relationships between set members. To qualify for basis set membership, shapes must satisfy certain constraints and limits on the size of such sets have been reported [5]–[8]. Basis decomposition can also be used to represent linear averaging and order-statistic filters [6], [8]. The basis representation is a result of theoretical importance and practical utility but, although its existence is well known, systematic general algorithms to generate the basis sets have not been available until now.

B. Scope and Organisation of this Work

The basis decomposition of morphologic operations provides three advantages:

1) full exploitation of the inherent parallelism in the morphologic processing of images;
2) a complete data-level description of the effects an operation has on local pixel distributions in any image;
3) the capability to distill the effect of many small data level changes into a single operation.

This paper provides a theoretical formalism and practical algorithms to find the basis set for any gray-scale morphological operation that has an algebra consisting of four operators: supremum, infimum, erosion and dilation. The results obtained apply to a broad range of useful operations. Basis decomposition of binary morphological operations is obtained as a particular case of the gray-scale theory.

Basic morphological operations, the kernel and the basis representations are defined in Section II.

Our general basis decomposition result is presented through the development of two tools. The first tool, given in Section III, generates the basis set for a serial suprema of erosions. The second tool, given in Section IV, shows how to generate, for any supremum of erosions, the equivalent (dual) set of basis filters expressed as an infimum of dilations, and vice-versa. This result enables these operations to be interchanged, so that our result for a serial suprema of erosions can be applied more generally.

Section V combines the tools developed in Sections III and IV and applies the methods developed to decompose the generalised open and close operations. Section VI is a summary of the work presented and surveys practical applications and theoretical extensions of the methods developed.

C. Basis Versus Structural Decomposition

Basis decomposition differs from the approach of structural decomposition of structuring elements [10], [11]. Structural decomposition explores alternative representations of a shape as an equivalent sequence of dilations by smaller geometric sub-shapes and seeks to find minimal expressions to achieve this. Structural decomposition simplifies the implementation of structuring elements with large area masks by enabling a series of masks with smaller regions of support, compatible with common hardware capacity, to be used. The shapes produced by structural decomposition result more from hardware restrictions rather than geometric considerations.

II. Definitions

A. Gray-Scale Operations

A gray-scale image may be considered as a function which maps pixel coordinates to gray-scale values. A structuring element, which is similarly defined, interacts with the image to modify it. For a gray-scale image f and structuring element g the erosion and dilation operations are defined respectively as

$$(f \ominus g)(x) = \inf_{z \in G, (x+z) \in F} \{f(x+z) - g(z)\} \quad (1)$$

$$(f \oplus g)(x) = \sup_{z \in G, (x-z) \in F} \{f(x-z) + g(z)\} \quad (2)$$

where $f : F \to E$, $g : G \to E$ and $F, G \subseteq E^{N-1}$ [1].

In this paper we work in Euclidean N-space and the infimum (inf) and supremum (sup) operators are used throughout. If the underlying space is discrete these operations may be replaced by the minimum and maximum operations respectively. We adopt the convention that a function has a value of negative infinity outside the region of its support. This simplifies the use of the infimum and supremum operators over functions with different support.

We assume that any structuring element $g : G \to E$ has a support G which is a *finite* set. This assumption is a necessary requirement for the basis sets to have a finite number of members.

B. The Kernel Representation

Morphological operations which can be represented as a supremum of erosions must have two important properties.

1) The Operation is Translation-Invariant (TI): An operation ψ on a gray-scale image f is TI iff:

$$\psi(f_y + c) = [\psi(f)]_y + c, \quad \forall f, \forall (y, c) \in F \times E,$$

where $f_y(x) = f(x - y)$ is the translation of f along y, and c is a translation of the amplitude of f.

2) The Operation is Increasing: An operation ψ on gray-scale images f and g is increasing iff:

$$f \leq g \Rightarrow \psi(f) \leq \psi(g), \quad \forall f, g.$$

Erosion, dilation, open and close are examples of TI increasing operations. If any operation ψ is TI increasing then it can be equivalently expressed as a supremum of erosions, such that [5]–[7]

$$\psi(f) = \sup_i f \ominus g_i. \quad (3)$$

The set of filters $\{g_i\}$ directly depend on the operation ψ performed on the image, but not on the image f. A constructive method to find all the members of the filter set for any given operation ψ is required. The kernel $K(\psi)$ of the operation ψ is defined to be

$$K(\psi) = \{g_i : [\psi(g_i)](\vec{0}) \geq 0\}. \quad (4)$$

The kernel contains the sets of filters $\{g_i\}$ that satisfy (3). The set of filters $\{g_i\}$ may be interpreted as images or data structures in an image. The constraint on g_i in (4) can be seen to arise by replacing f with g_i in (3), as equation (3) must hold for all images f, including the image of g_i.

Unfortunately, the kernel cannot be applied practically as it contains an infinite number of members. For any $g_i \in K(\psi)$ there are an infinite number of functions $g_j \geq g_i$ (called *superfunctions* of g_i) that will also be members of the kernel. However, superfunctions are redundant when used in a supremum of erosions: If $g_2, g_1 \in \{g_i\}$ where g_2 is a superfunction of g_1 then $g_2 \geq g_1 \Rightarrow f \ominus g_2 \leq f \ominus g_1$ and hence the supremum of $f \ominus g_2$ and $f \ominus g_1$ is always $f \ominus g_1$. Similarly, $g_2 \geq g_1 \Rightarrow g_2$ is redundant in an infimum of dilations.

The basis $B(\psi)$ contains only the non-redundant members of the kernel, and is defined [6] as

$$B(\psi) = \{g_i \in K(\psi) : [g_j \in K(\psi), \quad g_j \leq g_i] \Rightarrow g_j = g_i\}. \quad (5)$$

Equation (3) can now be written exactly as

$$\psi(f) = \sup_{g_i \in B(\psi)} f \ominus g_i. \quad (6)$$

The set of filters $\{g_i\}$ in (6), which is referred to as a basis set, is the minimal set of filters that represents directly the effects of the operation ψ on an image. The basis set can be analysed to explore the exact properties of a given operation. The effects of different morphological operations can be viewed on common ground when each is expressed as a supremum of erosions, as the various properties of different operations are manifest as differences in the basis sets.

III. Serial Erosions as Single Erosions

In this section the first part of our general basis decomposition result is established by computing the decomposition set for a serial suprema of erosions. Consider an operation ψ that is a cascade of two operations θ and ϕ, where

$$\theta(f) = \sup_m f \ominus a_m,$$
$$\phi(f) = \sup_n f \ominus b_n$$

and

$$\psi(f) = \phi(\theta(f)). \tag{7}$$

The operation ψ has a basis set which depends on the basis sets for θ and ϕ. A procedure to find the basis set for ψ can be applied to any operation that is a cascade of two TI increasing operations. The open-close, close-open operations and two passes of a median filter are examples of such operations. Furthermore, this procedure can be applied iteratively to allow the decomposition of arbitrarily long cascades of TI increasing operations.

A. A Parallel Expression for Serial Erosions

The following theorem provides the decomposition set for a serial suprema of erosions in terms of the filter sets $\{a_m\}$ and $\{b_n\}$. The proof appears in the Appendix.

Theorem 1: Let f be a gray-scale image and $\{a_m\}$ and $\{b_n\}$ be two sets of gray-scale filters. Let $B_n = \{z_{n,i} : i = 1, \cdots, \chi(B_n)\}$ be the support of the function $b_n : B_n \to E$, where $B_n \subseteq E^{N-1}$, and $\chi(B_n)$ is the cardinality of B_n. Define the pulse functions $b_{n,i} : \{z_{n,i}\} \to E$ to map a single coordinate $z_{n,i} \in B_n$ to a gray-scale value $b_{n,i}(z_{n,i}) = b_n(z_{n,i})$. Then,

$$\sup_n(\sup_m f \ominus a_m) \ominus b_n = \sup_{n,m_1,\cdots,m_{\chi(B_n)}} f \ominus c_{n,m_1,\cdots,m_{\chi(B_n)}}$$

where

$$c_{n,m_1,\cdots,m_{\chi(B_n)}} = \sup_i b_{n,i} \oplus a_{m_i}. \tag{8}$$

Note that the filter sets $\{a_m\}$ and $\{b_n\}$ consist of a finite number of filters, each with finite support, and that therefore the number of filters in the resultant decomposition set (8) is finite. However, it is not a basis set as it is possible for members of the set to be superfunctions of other members. Theorem 1 provides a finite decomposition set for any serial suprema of erosions which subsequently can be sorted for non-redundant members to obtain the basis set. The level of redundancy is not predicable as it is a function of the structural overlap of the (arbitrary) a_m and b_n filters. The possibility of redundancy arises because all possible filter combinations are allowed and must be considered.

B. An Algorithm to Compute the Basis Set for Serial Erosions

The following algorithm can be used to compute the basis members of the decomposition set $\{c_{n,m_1,\cdots,m_{\chi(B_n)}}\}$ from the given filter sets $\{a_m\}$ and $\{b_n\}$.

Algorithm 1:

Step 1: Select a filter b from the set $\{b_n\}$. Define $\chi(B)$ as the cardinality of B (i.e $\chi(B)$ is the number of pulse functions b_i that make up b).

Step 2: Choose a filter a_{m_i} from $\{a_m\}$ for each of the pulse functions b_i.

Step 3: Form the gray-scale dilation $b_i \oplus a_{m_i}$ using equation (2) for each b_i.

Step 4: The filter $c_{m_1,\cdots,m_{\chi(B)}}$ is formed by taking the supremum of all these dilations. For a particular coordinate $(\vec{x})$ the supremum result of these dilations at $(\vec{x})$ is taken as the value for $c_{m_1,\cdots,m_{\chi(B)}}(\vec{x})$.

Step 5: Repeat STEP 2 to STEP 4 for every possible association of a_{m_i} with the pulse functions b_i.

Step 6: Repeat STEP 1 to STEP 5 for each of the filters b_n in the filter list $\{b_n\}$ to form the complete set of decomposition filters $\{c_{n,m_1,\cdots,m_{\chi(B_n)}}\}$.

Step 7: Remove the redundant filters from the set $\{c_{n,m_1,\cdots,m_{\chi(B_n)}}\}$ to form the basis set. A filter c_R is redundant if $c_R(\vec{x}) \geq c(\vec{x})$, for all $(\vec{x})$ in the support of c, where c is another member of the filter list.

If M is the number of members in the basis set $\{a_m\}$ and N is the number of members in $\{b_n\}$ then the total number of members in the decomposition set $\{c_{n,m_1,\cdots,m_{\chi(B_n)}}\}$ is given by $C = \sum_{n=1}^{N} M^{\chi(B_n)}$. Usually many of these members will be redundant however and will not be members of the final basis set. How to predict the number of members in the final basis set is not known but it will always have an upper bound of C. Assuming that the time necessary to compute each dilation $b_i \oplus a_{m_i}$ is a constant k for all associations of b_i and a_{m_i} then the total amount of time necessary to compute the decomposition set is $k \times C$. Of course, C will increase rapidly as the size of the input sets increase and this may eventually lead to computational difficulties. Here the amount of memory required to store the decomposition set before it is reduced to the basis set is of more concern than the total computation time, as the time constant k for each computed dilation is very small. In practice the storage problem is mitigated by removing redundant members as they are computed (by comparing them to the existing basis elements that have already been computed). In such a way only members of the basis set are stored in memory at any given time.

As an example, consider the set of three (flat) filters $\{g_i\}$ in Fig. 1(a). A supremum of erosions using these three filters is equivalent to a 1-D median filter with a window $W = \{(-1,0),(0,0),(1,0)\}$ [8]. Figures 1(b) and 1(c) illustrate the basis sets that represent two and three passes of this filter respectively. The set $\{h_i\}$ in Fig 1(b) was computed using Algorithm 1 with $\{a_m\} = \{b_n\} = \{g_i\}$ and consists of 5 basis members derived from a decomposition set of $C = 27$ members. The set in Fig. 1(c) was computed using $\{a_m\} = \{h_i\}$ and $\{b_n\} = \{g_i\}$ and consists of 7 basis members derived from a decomposition set of $C = 75$ members. The procedure can continue in this manner to decompose an arbitrary number of applications of the filter.

IV. Mapping to the Dual Basis Set

A. The Dual Representation

Any supremum of erosions can be equivalently expressed as an infimum of dilations using the dual filter set. To provide greater flexibility in the decomposition of morphological op-

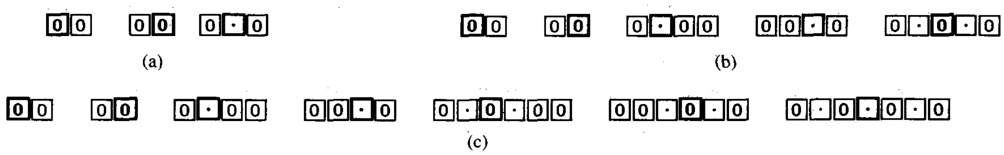


Fig. 1. Basis sets for a 1D median filter defined in a window $W = \{(-1,0),(0,0),(1,0)\}$. (a) The basis set $\{g_i\}$ for one pass of the filter. (b) The basis set $\{h_i\}$ for two passes. (c) The basis set for three passes. The origin is marked by the box in bold type. The number "0" indicates that that coordinate has a gray-scale of zero, the "." sign indicates the filter is undefined at that coordinate (and has a value $-\infty$).

erations, a constructive algorithm establishing a link between dual filter sets is required, in order that one set may be calculated from the other. The following theorem establishes such a relationship (the proof appears in the Appendix).

Theorem 2: Let f be a gray-scale image and $\{g_n\}$ be a set of gray-scale filters. Let $G_n = \{z_{n,i} : i = 1, \cdots, \text{card}(G_n)\}$ be the support of the function $g_n : G_n \to E$, where $G_n \subseteq E^{N-1}$. Define the pulse functions $g_{n,i} : \{z_{n,i}\} \to E$ to map a single coordinate $z_{n,i} \in G_n$ to a gray-scale value $g_{n,i}(z_{n,i}) = g_n(z_{n,i})$. Define the reflection of a function $g_{n,i}$ as $\breve{g}_{n,i}(z) = g_{n,i}(-z)$, and define N as the number of g_n filters. Then,

$$\inf_n f \oplus g_n = \sup_{i_1,\cdots,i_N} f \ominus k_{i_1,\cdots,i_N}$$

where

$$k_{i_1,\cdots,i_N} = \sup_n(-\breve{g}_{n,i_n}).$$

Conversely, from a given set $\{g_n\}$ used in a supremum of erosions, the set $\{k_{i_1,\cdots,i_N}\}$ required for an equivalent infimum of dilations operation can be computed. The proof deriving $\{k_{i_1,\cdots,i_N}\}$ is similar to Theorem 2, and the result is stated without derivation in the following theorem.

Theorem 3: Using the definitions of Theorem 2,

$$\sup_n f \ominus g_n = \inf_{i_1,\cdots,i_N} f \oplus k_{i_1,\cdots,i_N}.$$

Note that the filter set $\{k_{i_1,\cdots,i_N}\}$ in Theorem 3 is identical to that in Theorem 2.

B. An Algorithm to Compute the Dual Basis Set

The following algorithm can be used to compute the basis members of the dual filter set $\{k_{i_1,\cdots,i_N}\}$ from a given filter set $\{g_n\}$.

Algorithm 2:

Step 1: Transform the set $\{g_n\}$ to the set $\{-\breve{g}_n\}$, where $-\breve{g}_n(z) = -g_n(-z)$.

Step 2: Select one pulse function $-\breve{g}_{n,i}$ from each of the filters $-\breve{g}_n$

Step 3: A filter $k_{i_1,\cdots,i_N}$ is formed by combining all the selected pulse functions into one filter. If several pulse functions have the same coordinate but different gray-scales then the supremum gray-scale is chosen for that coordinate.

Step 4: Repeat Steps 2 and 3, selecting every possible combination of pulse functions.

Step 5: Remove redundant filters as in Step 7 of Algorithm 1.

If N is the number of members in $\{g_n\}$ and $\chi(G_n)$ is the cardinality of G_n (i.e. the number of pulse functions that make up g_n) then there are $\Pi_{n=1}^{N}\chi(G_n)$ members in the decomposition set $\{k_{i_1,\cdots,i_N}\}$. The number of elements in the final basis set will always be less than this upper bound, though we have not been able to predict an exact figure. The computation time for the algorithm is also directly proportional to this factor. Each filter $k_{i_1,\cdots,i_N}$ will consist of N pulse functions (one from each filter $-\breve{g}_n$) but there is often overlap between each $-\breve{g}_n$ and so the number of pulse functions in $k_{i_1,\cdots,i_N}$ is usually less than N. In practice any computational difficulties that may arise for large input sets can be offset by removing redundant members as they are computed.

Note that if the dual filter set $\{k_{i_1,\cdots,i_N}\}$ is used as the input to Algorithm 2, the resultant output will be the original set $\{g_n\}$. The algorithm will be used in the following section to decompose the open and close operations.

V. Decomposition of the Open and Close Operations

The dual morphological operations of opening and closing are serial operations defined respectively as

$$f \circ g = (f \ominus g) \oplus g \tag{9}$$

$$f \bullet g = (f \oplus g) \ominus g. \tag{10}$$

These operations can be applied in a more general form using multiple structuring elements. For example, Song and Delp used multiple structuring elements to remove impulse noise in images [12], and Stevenson and Arce used multiple structuring elements in a line preserving filter [13]. The operations were applied as a supremum of openings and an infimum of closings. Both operations can be decomposed into basis filters and have a basis representation

$$\sup_n f \circ g_n = \sup_i f \ominus k_i = \inf_i f \oplus k^i \tag{11}$$

$$\inf_n f \bullet g_n = \sup_i f \ominus k^i = \inf_i f \oplus k_i. \tag{12}$$

The basis sets $\{k_i\}$ and $\{k^i\}$ in equation (11) are identical to those in (12), reflecting the duality of the open and close operations as demonstrated in [5]–[7].

Often the effects of one operation will be provided partially by another operation [14], [21]. Combining multiple operations will be inefficient if the effects of the constituent operations overlap. The basis set for the combined operation is the minimal description of the operation, representing any overlapping properties only once. It can have *less* members than the basis sets for the constituent operations. An example demonstrating this is given in Section B.

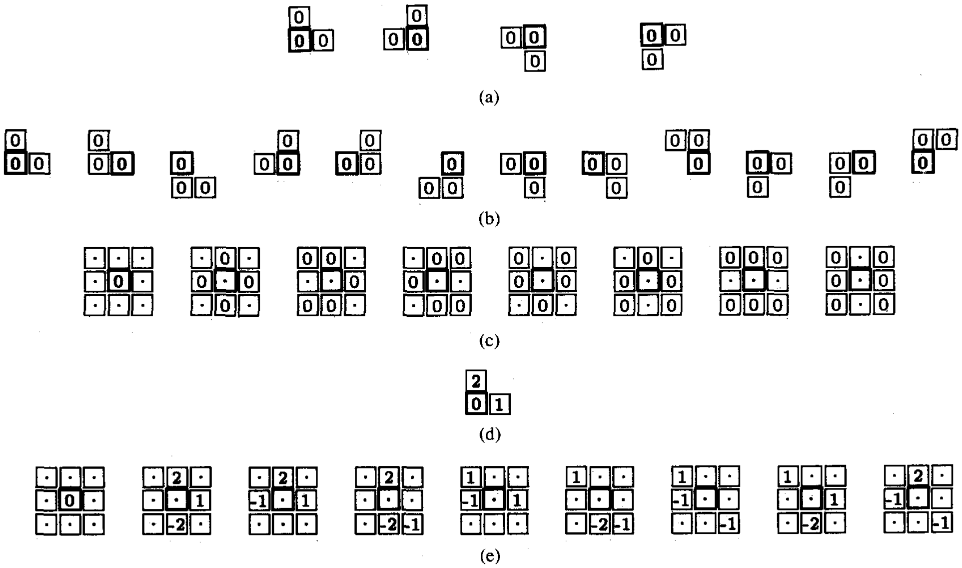


Fig. 2. Basis sets for the multiple open and close operations. (a) A set of four structuring elements. (b) The basis set for opening. (c) The basis set for closing. (d) A single "L" type structuring element with gray-scale weights. (e) The basis set for closing. The origin is marked by the box in bold type. The numbers in the boxes indicate gray-scale values at coordinates. The "." sign indicates that the filter is undefined at that coordinate.

A. Basis Sets for the Open and Close Operations

The multiple open operation (11) can be decomposed into basis filters as follows. Each open is an erosion followed by a dilation (9). The dilation can be decomposed into a supremum of erosions using Algorithm 2. Each open is then a serial suprema of erosions, and by using Algorithm 1 can be expressed as a single supremum of erosions. Each supremum of erosions, corresponding to each open, can then be grouped under the initial supremum operator to form a single supremum of erosions.

An explicit expression for the decomposition set for opening operation is given below (proof in Appendix).

Corollary 1: Using the definitions of Theorem 2,

$$\sup_n f \circ g_n = \sup_{n,i} f \ominus k_{n,i}$$

where

$$k_{n,i} = (-\breve{g}_{n,i}) \oplus g_n.$$

The decomposition set for the multiple close operation (12) can be obtained from a similar procedure as that outlined above for the opening. An explicit expression for the decomposition set for closing is given in the Appendix.

B. An Algorithm to Compute the Open and Close Basis Sets

The duality between the basis sets in (11) and (12) indicates that given the basis set for opening $\{k_i\}$ in (11), the basis set for closing $\{k^i\}$ in (12) can be computed using Algorithm 2, and vice versa. The following algorithm can be used to compute the basis sets for opening and closing, given a set of structuring elements $\{g_n\}$. Note that we use the open and close duality by computing the basis set for closing from the basis set for opening, through Algorithm 2.

Algorithm 3:

- Step 1: Select a structuring element g from the set $\{g_n\}$.
- Step 2: Transform g to $-\breve{g}$.
- Step 3: Select a pulse function $-\breve{g}_i$ from $-\breve{g}$. A basis filter for the open operation is computed by forming the gray-scale dilation $(-\breve{g}_i) \oplus g$ using equation (2).
- Step 4: Repeat Step 3 for every pulse function $-\breve{g}_i$ in $-\breve{g}$.
- Step 5: Repeat Step 1 to Step 4 for every structuring element g_n to complete the basis set for opening.
- Step 6: The basis set for closing is computed using Algorithm 2, taking the basis set for opening computed above as the input set $\{g_n\}$.

If N is the number of structuring elements in the set $\{g_n\}$ and $\chi(G_n)$ is the number of pulse functions that make up g_n then the total number of elements in the decomposition set for opening is given by $\sum_{n=1}^{N} \chi(G_n)$. The time taken to compute the set is directly proportional to this factor. Clearly this will not pose any computational problems even for very large input sets. The basis set for closing has parameters that have already been associated with Algorithm 2 for computing the dual basis.

Fig. 2(a) shows a series of four "L" type structuring elements (as used by Song and Delp [12]). The structuring elements are to be used in the multiple open (11) and close (12) operations. Illustrated in Figs. 2(b) and (c) are the basis sets for opening and closing respectively. The basis set for the multiple opening has twelve members, and therefore the parallel implementation of the opening would be easier as an infimum of dilations using the eight members of the basis set for closing. The basis implementation (eight erosions in parallel for the closing) competes favourably with the standard serial implementation (four dilations followed by four erosions in parallel for the closing).

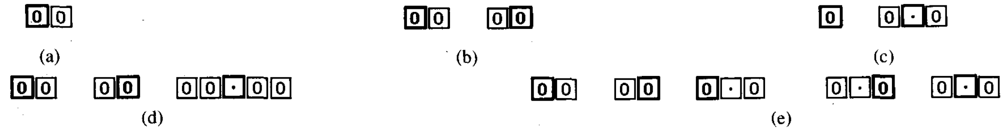


Fig. 3. Basis sets for the open-close and close-open operations. (a) A structuring element with coordinates $\{(0,0),(1,0)\}$. (b) The basis set for opening. (c) The basis set for closing. (d) The basis set for the close-open operation. (e) The basis set for the open-close operation. The origin is marked in bold type. Numbers indicate gray-scale values at those coordinates, the "." sign indicates that the filter is undefined at that coordinate.

In Fig. 2(d) is one of the "L" type structuring elements with arbitrary gray-scale weights. The corresponding basis set for closing is shown in Fig. 2(e). The basis set for this gray-scale structuring element has nine members, whereas the basis set for the multiple closing has only eight members. The number of basis filters is less due to the overlapping support of the four structuring elements. Note that the basis set for a single flat "L" also has nine members, although in general the choice of gray-scale weights will affect the number of filters in the resultant basis set. The number of basis filters resulting from gray-scale decomposition is always equal to or greater than that for flat structuring elements, due to the extra degree of freedom afforded by the gray-scale weights.

VI. Applications, Extensions, and Future Work

A. Applications of the Basis Representation

Parallel Implementation: Executing image transformations as parallel operations has the advantage of maximising processing speed and is attractive as it conforms to plausible models of the human visual system. Parallel expressions often give a clearer insight into the detailed workings of an operation. Expressing cascaded morphological operations as a single supremum of erosions removes the need to visualize intermediate steps and provides a template-like filtering process. The disadvantage of parallel processing lies in the time-for-memory-trade-off; complex sequential operations generate many valid permutations of data, with potentially large support, which need to be tested in one pass.

In binary morphology, a basis can be treated as a set of binary templates which target data structures in the image that match the templates. They can then be placed into a look-up table (LUT) to effect a parallel implementation of the operation [15]. Consider for example the set of flat basis filters illustrated in Fig. 2(c). These eight basis filters all lie within a 3 by 3 region of support, in which there can form $2^{(3\times 3)} = 512$ possible binary data patterns. Each basis filter forms its own binary data pattern which is coded as an address vector indexing a 512 entry LUT. The LUT is set to ON for image patterns that match the pattern of a basis filter, and OFF for other patterns. In total the implementation requires a 3 by 3 convolution (with kernel weights 2^i, $0 \leq i \leq 8$) to obtain the address vector for each data pattern, followed by a point-wise mapping through the LUT. For regions of support that are larger than 3 by 3, overlapping 512 byte LUT's can be used [15]. The use of LUT's, made possible by the basis representation, is particularly attractive when multiple structuring elements are being used, as the standard implementation of multiple structuring elements is more complex.

Data-Level Description: It has been shown that the basis representation can simplify morphological operations that use multiple structuring elements. The basis set also allows direct comparison of distinct but complementary operations. For example, the open-close and close-open operations are known [8] to bound the original image and the output from the median filter. The detailed differences between these operations is evident in their basis decomposition sets.

Illustrated in Fig. 3(a) is a flat 2 by 1 structuring element with coordinates $\{(0,0),(0,1)\}$. The opening and closing basis sets for this structuring element, computed using Algorithm 3, are illustrated in Figs. 3(b) and 3(c) respectively. Using $\{a_m\}$ as the basis set for opening and $\{b_n\}$ as the basis set for closing, Algorithm 1 can be used to compute the basis set for the close-open operation, as shown in Fig. 3(d). In a similar way the open-close operation can be decomposed. That basis set is shown in Fig. 3(e).

The basis representation allows visual comparison between the detailed effects of disparate operations. It offers a unified description which reveals relationships that may be difficult to establish theoretically. For example, the basis sets reveal the relative extensive and antiextensive properties of operations. Every filter in Fig. 3(d) is a superfunction of some filter in Fig. 3(e). As $g_1 \geq g_2 \Rightarrow f \ominus g_1 \leq f \ominus g_2$ this implies that a supremum of erosions with the filter set in 3(d) is contained in a supremum of erosions with the filter set in 3(e). For this particular example, $(f \circ g) \bullet g \leq (f \bullet g) \circ g$ (this relation does not hold for an arbitrary structuring element). By considering the basis sets in Figures 3(b) and 3(c), the relation may be extended to $f \circ g \leq (f \circ g) \bullet g \leq (f \bullet g) \circ g \leq f \bullet g$.

These basis sets may be compared to that for the three point median filter in Fig. 1. If $\text{med}_n(f)$ denotes n passes of the median filter, it is apparent that $(f \circ g) \bullet g \leq \text{med}_1(f) \leq (f \bullet g) \circ g$, $(f \circ g) \bullet g \leq \text{med}_2(f) \leq (f \bullet g) \circ g$, and that $(f \circ g) \bullet g \leq \text{med}_3(f) \leq (f \bullet g) \circ g$. This visually illustrates that the median filter in 1D is bounded by the open-close and close-open operations [8]. Note that the above conclusions are independent of the image f but are particular to the structuring element chosen.

Tailoring Filters: By first identifying the structures in an image that need to be changed or preserved, it is possible to construct from this chosen set a serial operation that best approximates the required result. Because adding or subtracting individual basis filters makes only small differences to the total final result, very fine control over the filtering process is possible. This approach is preferable to choosing additional structuring elements to achieve the selectivity required.

Ultimately, the degree of change in a filtered image depends on the image; basis decomposition allows a constructive mechanism to fine tune the filter characteristics in an image independent way.

B. Extensions and Future Work

Basis Shape Properties: We are examining the basis sets given systematic changes to two- and three-dimensional structuring elements on rectangular and hexagonal lattices, to characterise the growth of the basis sets and the grouping of common types of data structures within the basis sets.

The algorithms we have presented involve the computation of a finite decomposition set for an operation which can subsequently be sorted to form the basis set. Redundancy within basis sets becomes an important issue when there are a large number of multiple structuring elements, or when the structuring elements have a large support, as the number of basis filters that must be selected and sorted for redundancy diverges rapidly. We are investigating algorithms that will select basis filters directly.

Idempotent Basis Sets: Idempotence is an attractive property of some morphological operations (such as open and close). Successive applications of such operations leave an image unchanged, that is, all possible changes to an image occur after one application of the operation. If an operation $\psi(f) = \sup_n f \ominus g_n$ is an idempotent TI increasing operation, the idempotent property is explicitly represented by the basis filters, and is expressed by the relation

$$\sup_n(\sup_n f \ominus g_n) \ominus g_n \equiv \sup_n f \ominus g_n. \tag{13}$$

For this equation to hold, Theorem 1 indicates that the set of filters $\{g_n\}$ must be of the form as given by equation (8). The circumstances under which a general supremum of erosions becomes idempotent are of interest with respect to the design of idempotent morphological operations.

Structuring Element Synthesis: Isolated data structures that resemble close basis filters are good candidates from which representative filters can be synthesized. The mechanism to synthesize structuring elements from basis filters usually involves the conversion of the basis set for the close operation to the dual open set. Because members of the open set are coherent translates of the structuring element, it is possible to identify compact structures that represent the initial set of structures taken from the data. We need to refine ways to select appropriate pools of input data filters and to improve the translate matching process to help extract these representative structuring elements.

The tools provided in this work can be used to obtain the basis decomposition for a wide variety of morphological operations. They have application to the design of selective filters and to evaluate the feasibility of parallel implementation. The basis representation should play a more significant role in the design of morphological operations, as the detailed differences in the operations are clearly manifest as differences in the basis filters. Such an approach should be used to avoid the heuristic choices adopted when deciding which operation, and which structuring elements, should be used to process images.

ACKNOWLEDGMENT

We thank the anonymous referees for their useful comments and suggestions. R. Jones would like to thank O. Jones for his support and help in the preparation of this paper.

APPENDIX

The following transform properties will be used in the proofs of the theorems in this paper.

$$f \oplus g = g \oplus f \tag{A1}$$
$$(f \ominus g) \ominus k = f \ominus (g \oplus k) \tag{A2}$$
$$f \oplus (\sup_i g_i) = \sup_i f \oplus g_i \tag{A3}$$
$$f \ominus (\sup_i g_i) = \inf_i f \ominus g_i \tag{A4}$$
$$\inf_i(\sup_j f_{i,j}) = \sup_{j_1,\cdots,j_I}(\inf_i f_{i,j_i}) \tag{A5}$$

The following two properties concern a *pulse function* p that maps a single element of E^{N-1} to E

$$f \oplus p = f \ominus (-\check{p}) \tag{A6}$$
$$(\sup_i f_i) \ominus p = \sup_i (f_i \ominus p). \tag{A7}$$

Proof of Theorem 1:

$$\begin{aligned}
&\sup_n(\sup_m f \ominus a_m) \ominus b_n \\
&= \sup_n[(\sup_m f \ominus a_m) \ominus (\sup_i b_{n,i})] && \text{(by defn.)} \\
&= \sup_n[\inf_i\{(\sup_m f \ominus a_m) \ominus b_{n,i}\}] && \text{[A4]} \\
&= \sup_n[\inf_i\{\sup_m((f \ominus a_m) \ominus b_{n,i})\}] && \text{[A7]} \\
&= \sup_n[\sup_{m_1,\cdots,m_{\chi(B_n)}}\{\inf_i((f \ominus a_{m_i}) \ominus b_{n,i})\}] && \text{[A5]} \\
&= \sup_{n,m_1,\cdots,m_{\chi(B_n)}}\{\inf_i((f \ominus a_{m_i}) \ominus b_{n,i})\} \\
&= \sup_{n,m_1,\cdots,m_{\chi(B_n)}}\{\inf_i(f \ominus (a_{m_i} \oplus b_{n,i}))\} && \text{[A2]} \\
&= \sup_{n,m_1,\cdots,m_{\chi(B_n)}} f \ominus (\sup_i a_{m_i} \oplus b_{n,i}) && \text{[A4]} \\
&= \sup_{n,m_1,\cdots,m_{\chi(B_n)}} f \ominus (\sup_i b_{n,i} \oplus a_{m_i}) && \text{[A1]} \\
&= \sup_{n,m_1,\cdots,m_{\chi(B_n)}} f \ominus c_{n,m_1,\cdots,m_{\chi(B_n)}} && \text{(by defn.).}
\end{aligned}$$

Proof of Theorem 2:

$$\begin{aligned}
\inf_n f \oplus g_n &= \inf_n f \oplus (\sup_i g_{n,i}) && \text{(by defn.)} \\
&= \inf_n(\sup_i f \oplus (g_{n,i})) && \text{[A3]} \\
&= \inf_n(\sup_i f \ominus (-\check{g}_{n,i})) && \text{[A6]} \\
&= \sup_{i_1,\cdots,i_N}(\inf_n f \ominus (-\check{g}_{n,i_n})) && \text{[A5]} \\
&= \sup_{i_1,\cdots,i_N} f \ominus (\sup_n(-\check{g}_{n,i_n})) && \text{[A4]} \\
&= \sup_{i_1,\cdots,i_N} f \ominus k_{i_1,\cdots,i_N} && \text{(by defn.).}
\end{aligned}$$

Proof of Corollary 1:

$$\begin{aligned}\sup_n f \circ g_n &= \sup_n (f \ominus g_n) \oplus g_n \qquad \text{(by defn.)}\\ &= \sup_n [\sup_i ((f \ominus g_n) \ominus (-\breve{g}_{n,i}))] \\ &\qquad \text{(by Thm. 2)}\\ &= \sup_n [\sup_i f \ominus ((-\breve{g}_{n,i}) \oplus g_n)] \\ &\qquad \text{(by Thm. 1)}\\ &= \sup_{n,i} f \ominus ((-\breve{g}_{n,i}) \oplus g_n)\\ &= \sup_{n,i} f \ominus k_{n,i} \qquad \text{(by defn.).}\end{aligned}$$

Corollary 2: The decomposition set for the multiple close operation. Using the definitions of Theorem 2, and defining $\chi(G_n)$ as the cardinality of G_n,

$$\inf_n f \bullet g_n = \sup_{j_{1,1},\cdots,j_{1,\chi(G_1)},\cdots,j_{N,1},\cdots,j_{N,\chi(G_N)}} f \ominus k_{j_{1,1},\cdots,j_{1,\chi(G_1)},\cdots,j_{N,1},\cdots,j_{N,\chi(G_N)}}$$

where

$$k_{j_{1,1},\cdots,j_{1,\chi(G_1)},\cdots,j_{N,1},\cdots,j_{N,\chi(G_N)}} = \sup_{n,i} g_{n,i} \oplus (-\breve{g}_{n,j_{n,i}}).$$

Proof:

$$\begin{aligned}&\inf_n f \bullet g_n\\ &= \inf_n [(f \oplus g_n) \ominus g_n] \qquad \text{(by defn.)}\\ &= \inf_n [(\sup_i f \ominus (-\breve{g}_{n,i})) \ominus g_n] \qquad \text{(by Thm. 2)}\\ &= \inf_n [\sup_{i_1,\cdots,i_{\chi(G_n)}} f \ominus (\sup_j g_{n,j} \oplus (-\breve{g}_{n,i_j}))]\\ &\qquad \text{(by Thm. 1)}\\ &= \inf_n [\inf_j f \oplus ((-\breve{g}_{n,j}) \oplus g_n)] \qquad \text{(by Thm. 3)}\\ &= \inf_{n,j} f \oplus ((-\breve{g}_{n,j})) \oplus g_n\\ &= \sup_{j_{1,1},\cdots,j_{1,\chi(G_1)},\cdots,j_{N,1},\cdots,j_{N,\chi(G_N)}} f \ominus (\sup_{n,i} g_{n,i} \oplus (-\breve{g}_{n,j_{n,i}})) \qquad \text{(by Thm. 2)}\\ &= \sup_{j_{1,1},\cdots,j_{1,\chi(G_1)},\cdots,j_{N,1},\cdots,j_{N,\chi(G_N)}} f \ominus k_{j_{1,1},\cdots,j_{1,\chi(G_1)},\cdots,j_{N,1},\cdots,j_{N,\chi(G_N)}} \qquad \text{(by defn.).}\end{aligned}$$

REFERENCES

[1] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, pp. 523–550, July 1987.

[2] J. Serra, *Image Analysis and Mathematical Morphology. Vol. 2: Theoretical Advances.* New York: Academic, 1988.

[3] H. Heijmans and C. Ronse, "The algebraic basis of mathematical morphology I. Dilations and erosions," *Comput. Vision, Graphics, Image Processing*, vol. 50, no. 3, pp. 245–295, June 1990.

[4] H. Heijmans, "Theoretical aspects of gray-level morphology," *IEEE Pattern Anal. Machine Intell.*, vol. 13, no. 6, pp. 568–582, June 1991.

[5] G. Matheron, *Random Sets and Integral Geometry.* New York: Wiley, 1975.

[6] P. Maragos, "A representation theory for morphological image and signal processing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 6, pp. 586–599, June 1989.

[7] C. R. Giardina and E. R. Dougherty, *Morphological Methods in Image and Signal Processing.* Englewood Cliffs, N.J: Prentice Hall, 1988.

[8] P. A. Maragos and R. W. Schafer, "Morphological filters—Part I and part II," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, no. 8, pp. 1153–1184, Aug. 1987.

[9] I. D. Svalbe, "The geometry of basis sets for morphologic closing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 12, pp. 1214–1224, Dec. 1991.

[10] F. Shih and O. R. Mitchel, "Decomposition of gray-scale morphological structuring elements," *Pattern Recogn.*, vol. 24, no. 3, pp. 195–203, 1991.

[11] P. D. Gader, "Separable decompositions and approximations of greyscale morphological templates," *Comput. Vision, Graphics, Image Processing: Image Understanding*, vol. 53, no. 3, pp. 288–296, May 1991.

[12] J. Song and E. J. Delp, "The analysis of morphological filters with multiple structuring elements," *Comput. Vision, Graphics, Image Processing*, vol. 50, no. 3, pp. 308–328, June 1990.

[13] R. L. Stevenson and G. R. Arce, "Morphological filters: Statistical and further syntactical properties," *IEEE Trans. Circuits Syst.*, vol. 34, no. 11, pp. 1292–1305, Nov. 1987.

[14] I. Svalbe and R. Jones, "The design of morphologic filters using multiple structuring elements. Part I: Openings and closings," *Pattern Recognit. Lett.*, no. 13, pp. 123–129, Feb. 1992.

[15] R. Jones and I. Svalbe, "Morphological filtering as template matching," to be published in *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, no. 4, pp. 438–443, Apr. 1994.

[16] E. R. Dougherty and R. Loce, "Efficient design strategies for the optimal binary digital morphological filter: Probabilities, constraints, and structuring element libraries," *Mathematical Morphology in Image Processing*, E. Dougherty, Ed. New York: Marcel Dekker, 1992.

[17] G. J. F. Banon and J. Barrera, "Minimal representations of translation-invariant set mappings by mathematical morphology," *SIAM J. Appl. Math.*, vol. 51, no. 6, Dec. 1991.

[18] G. J. F. Banon and J. Barrera, "A decomposition theorem in mathematical morphology," in *Proc. Int. Workshop on Mathematical Morphol.*, Barcelona, Spain, May 1993.

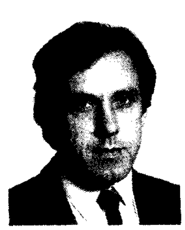
[19] M. Khosravi and R. W. Schafer, "A finite max-min representation for ILTI filters," *Proc. Int. Workshop on Mathematical Morphol.*, Barcelona, Spain, May 1993.

[20] C. Ronse and H. Heijmans, "The algebraic basis of mathematical morphology II. Openings and closings," *Comput. Vision Graphics, Image Processing: Image Understanding*, vol. 54, no. 1, pp. 74–97, July 1991.

[21] R. Jones and I. Svalbe, "The design of morphologic filters using multiple structuring elements. Part II: Open(close) and close(open)," *Pattern Recognit. Lett.*, no. 13, pp 175–181, Mar. 1992.

Ronald Jones was born in Wellington, New Zealand in June, 1968. He received the B.Sc.(Hons) degree in physics from Victoria University of Wellington in 1990. He is currently completing the Ph.D. degree in mathematical morphology at Monash University in Melbourne.

His research interests included basis representations, switching theory, and mathematical morphology in graphs.



Imants Svalbe (M'86) was born in Ballarat, Australia, in 1952. He received the B.Sc.(Hons) degree in physics in 1974 and the Ph.D. degree in experimental nuclear physics in 1978, both from the University of Melbourne.

He is currently a senior lecturer in the Department of Physics at Monash University. His principle interests are discrete geometry and digital measurement theory as applied to image transforms and low-level image processing.