Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

Basis Computation Algorithms

Nina S. T. Hirata, Roberto Hirata Jr. and Junior Barrera

Instituto de Matemática e Estatística (IME), Universidade de São Paulo (USP), SP, Brazil
{nina,hirata,jb}@ime.usp.br

Abstract Algorithms for the computation of the basis (maximal intervals of the kernel) of binary and gray-scale translation-invariant and locally defined morphological operators are presented. Some examples that illustrate the dynamics of the algorithms as well as their use as learning algorithms are also presented.

Keywords: kernel, basis, maximal intervals, incremental splitting of intervals.

1. Introduction

One interesting aspect of morphological operators is the existence of canonical representations [1, 2, 9–11]. According to the canonical decomposition theorem, any translation-invariant morphological operator can be expressed as a supremum of sup-generating (also known as interval) operators. Although this theorem holds for any translation-invariant mapping between two complete lattices, in this work we restrict the scope to binary and gray-level image operators. If we also impose local definition by a neighborhood W, it can be shown that all kernel elements can be restricted to W [5]. Such operators are called W-operators. A W-operator can be characterized by a function whose domain is restricted to sub-images in W and its value to gray-scales of the image. This fact is important if our concern is the computational representation of image operators.

In the binary case, a W-operator $\Psi : \mathcal{P}(E) \to \mathcal{P}(E)$, where $E = \mathbb{Z}^2$ denotes the image domain, is characterized by a binary function $\psi : \mathcal{P}(W) \to \{0, 1\}$. Its kernel is defined as

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

and its basis, denoted $\mathbf{B}(\Psi)$, consists of the set of maximal intervals in the kernel.

Interval operators are parameterized by a pair of structuring elements that form intervals. The kernel induces a canonical decomposition while the basis induces a minimal decomposition as a supremum of interval operators. In terms of its kernel and basis, ψ can be expressed respectively as follows:

$$\psi(X) = max\{\lambda_{[A,A]}(X) : [A, A] \subseteq \mathcal{K}(\psi)\}, \tag{2}$$

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

and

$$\psi(X) = max\{\lambda_{[A,B]}(X) : [A,B] \in \mathbf{B}(\psi)\}\,, \tag{3}$$

where the interval operator $\lambda_{[A,B]}$ is defined as

$$\lambda_{[A,B]}(X) = 1 \iff X \in [A,B]\,. \tag{4}$$

Similar definitions exist for gray-scale W-operators [3].

In this work we are particularly concerned with a specific problem. Suppose we have knowledge of some elements that are and that are not elements of a kernel. How do we compute a set of intervals that are consistent with this set of elements? In other words, elements known to be in the kernel must belong to at least one of the intervals and those known not to be in the kernel must not belong to any of the intervals. By representing the set of elements by intervals, our goal is to have a compact representation of the operator. In particular, we are interested in the representation of functions of the following types:

a) $\psi : \{0,1\}^n \to \{0,1\}$, $n > 0$ an integer,

b) $\psi : \{0,1\}^n \to \{0,1,2,\ldots,k\}$, $n, k > 0$ integers, and

c) $\psi : \{0,1,2,\ldots,k_1\}^n \to \{0,1,\ldots,k_2\}$, $n, k_1, k_2 > 0$ integers.

These functions characterize, respectively, W-operators (a) between binary images, (b) between binary and gray-level images, and (c) between gray-level images.

In the following three sections we present algorithms for the computation of a set of maximal intervals covering the elements that are known to be in the kernel and not covering the ones that are known not to be in the kernel, for the three cases above. In the last section we present the conclusions of this work.

2. Binary input and output

Finding maximal intervals of the kernel of a binary W-operator is equivalent to finding the minimal sum of products form of a Boolean function. Classical algorithms such as the one due to Quine and McCluskey are well known in the fields of switching theory and logic design [6]. It is known that this problem is NP (computationally intractable). Therefore many implementations use heuristics to circumvent computational difficulties. One of such implementations is the Berkeley Espresso [4].

While Quine-McCluskey's approach follows a bottom-up strategy of joining smaller intervals in the kernel in order to generate a larger one, until no more joining are possible, another alternative option we have proposed is based on a top-down approach [8]. The approach, called ISI (Incremental Splitting of Intervals), starts the process with the whole interval $[\emptyset, W]$ and then removes from this interval the elements that are not in the kernel. The

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

result of the removing operation can be expressed by means of a set of sub-intervals of the initial interval. Thus, subsequent removing have the effect of sequentially breaking (or splitting) one ore more of these intervals. At the end of the process, when all elements that are not in the kernel have been removed, only elements of the kernel will remain covered by the resulting intervals.

2.1 Algorithm review

The removing operation or splitting rule [8] can be thought as the difference between two intervals $[A, B]$ and $[C, D]$ expressed in terms of the maximal intervals contained in $[A, B] \setminus [C, D]$.

Proposition 1 (Splitting rule)**.** *Let $[A, B]$ and $[C, D]$ be two intervals in $[\emptyset, W]$. Then, the set of maximal intervals contained in $[A, B] \setminus [C, D]$ is given by*

$$\left\{[A, B \cap \{c\}^c] : c \in C \cap A^c\right\} \cup \left\{[A \cup \{d\}, B] : d \in D^c \cap B\right\}, \qquad (5)$$

where the complement $\cdot^c$ is with relation to W. If $C = D = X$, $X \in \mathcal{P}(W)$, the rule simplifies to

$$\{[A, B \cap \{a\}^c] : a \in X \cap A^c\} \cup \{[A \cup \{b\}, B] : b \in B \cap X^c\}. \qquad (6)$$

The ISI algorithm consists basically on applying the splitting rule repeatedly in order to remove from interval $[\emptyset, W]$ all elements X such that $\psi(X) = 0$ (those elements that are not in the kernel of ψ). Thus, after the removing process finishes, the remaining intervals cover the elements X such that $\psi(X) = 1$ and eventually some elements X such that $\psi(X)$ is unknown (don't cares). If $\{X : \psi(X) = 1\} \cup \{X : \psi(X) = 0\} = \mathcal{P}(W)$, then the resulting collection of intervals corresponds to the basis of ψ. Otherwise, it is a basis that is consistent with ψ with respect to the elements in $\{X : \psi(X) = 1\} \cup \{X : \psi(X) = 0\}$.

The collection of maximal intervals resulting from the removing process may contain redundant intervals. That is, it may contain an interval $[A, B]$ such that all its elements are contained in some other intervals of the collection (although $[A, B]$ itself is not contained in any other interval). Hence, in a second step of the ISI algorithm a minimum cover (sub-basis) should be computed [6]. A minimum cover corresponds to a smallest sub-collection of intervals that is enough to cover all elements in $\{X : \psi(X) = 1\}$.

When don't cares are present, some heuristics can be applied to accelerate the algorithm. For instance, during the removing process, intervals that do not contain any element of $\{X : \psi(X) = 1\}$ can be disregarded because they will not be needed for the minimum cover. As an example, in Figure 1, filled circles represent the elements of the kernel, non-filled circles are the elements for which ψ takes value 0, and the absence of a circle indicates an element for which the value of ψ is not defined (dont't care). Figure 1(a)

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

shows the removing of element 001 from the interval $[000, 111]$, resulting in three intervals, $[000, 110]$, $[100, 111]$ and $[010, 111]$. The one (enclosed by a dashed box) does not contain any element of $\mathcal{K}(\psi)$ and, therefore, it can be discarded, as illustrated in Figure 1(b).

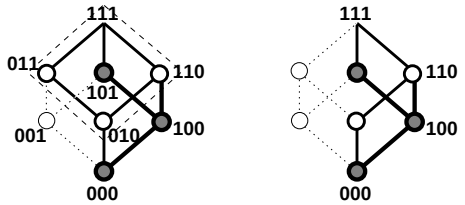


Figure 1. (a) The three intervals after the removing of 001. (b) Discarding of interval $[010, 111]$, which will not be needed for the minimum cover.

More details of this algorithm, as well as some interesting heuristics to improve the processing time, can be found in [8].

2.2 Application example

The above algorithm may be used as a learning (generalization) algorithm, as shown in the next example. Figure 2(a) shows an input-output pair of training images (to be used to estimate a kernel). From the training images,

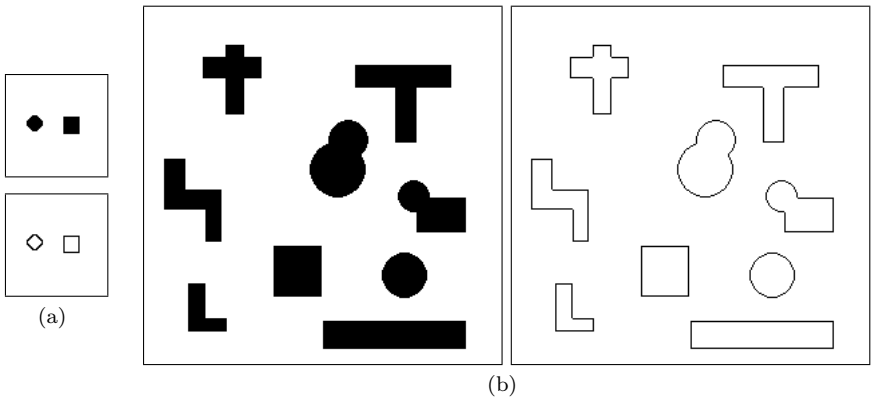
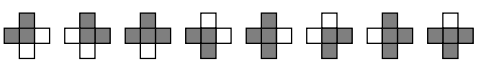


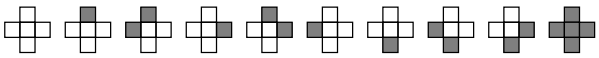
Figure 2. Learning binary image operators: (a) training images and (b) test images.

the elements known to be in the kernel are



Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

while those known not to be in the kernel are



The resulting basis is given by the set of intervals

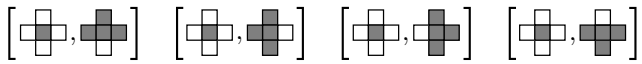


Figure 2(b) shows the result obtained by applying the above basis on another image. The learned operator is the 8-connected internal edge extractor. Additional application examples can be found in [7].

3. Binary input and multi-level output

Operators that map binary images to gray-scale images can be characterized by a function of the form $\psi : \{0,1\}^n \to \{0,1,\ldots,k\}$, where k denotes the maximum gray-level. Their kernel are defined by level, that is, the *kernel of ψ at level i*, $i = 0,1,\ldots,k$, is the collection $\mathcal{K}_i(\psi)$ given by, for any $\mathbf{x} \in \{0,1\}^n$,

$$\mathbf{x} \in \mathcal{K}_i(\psi) \iff \psi(\mathbf{x}) \geq i. \tag{7}$$

Notice that $\mathcal{K}_k(\psi) \subseteq \mathcal{K}_{k-1}(\psi) \subseteq \cdots \subseteq \mathcal{K}_1(\psi) \subseteq \mathcal{K}_0(\psi)$. The *basis of ψ at level i*, $\mathbf{B}_i(\psi)$, is the collection of maximal intervals in $\mathcal{K}_i(\psi)$.

The sup-decomposition [3] of ψ in terms of its kernel and basis are given, respectively, $\forall \mathbf{x} \in \{0,1\}^n$, by

$$\psi(\mathbf{x}) = max\{i : \mathbf{x} \in \mathcal{K}_i(\psi), i = 0,1,\ldots,k\}, \tag{8}$$

and

$$\psi(\mathbf{x}) = max\{i : \mathbf{x} \in [\mathbf{a},\mathbf{b}], [\mathbf{a},\mathbf{b}] \in \mathbf{B}_i(\psi), i = 0,1,\ldots,k\}\,. \tag{9}$$

3.1 Algorithm

To compute the basis $\mathbf{B}_i(\psi)$, $i = 0,1,\ldots,k$, of $\psi : \{0,1\}^n \to \{0,1,\ldots,k\}$, ISI is applied repeatedly, one time for each level i. Initially, all elements with label 0 are removed from the interval $[\emptyset, W]$. In this process, all other elements are regarded as 1s. The resulting intervals correspond to $\mathbf{B}_1(\psi)$, the intervals that cover all elements with label greater or equal to 1. In the next step, the same process is repeated for all elements with label 1. This time, the initial intervals are those in $\mathbf{B}_1(\psi)$ and the resulting intervals correspond to $\mathbf{B}_2(\psi)$. This process is repeated successively for the next labels until all elements of label $k-1$ are removed, resulting in the basis at level k.

As an example, consider $\psi : \mathcal{P}(W) \to \{0,1,2,3\}$, with $|W| = 3$, given by $\psi(000) = 0$, $\psi(010) = 1$, $\psi(100) = 1$, $\psi(101) = 2$, $\psi(110) = 2$ and $\psi(111) = 3$. The remaining elements are don't cares. Figure 3 shows the

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

dynamics of the algorithm, while Figure 4 shows the resulting basis. The basis at level 0, $\mathbf{B}_0(\psi)$, is the set composed by the interval $[\emptyset, W]$. To compute $\mathbf{B}_1(\psi)$, all elements with label 0 are removed from the interval $[\emptyset, W]$, resulting in $\mathbf{B}_1(\psi) = \{[001, 111], [010, 111], [100, 111]\}$. To compute $\mathbf{B}_2(\psi)$, all elements with label 1 are removed from the intervals in $\mathbf{B}_1(\psi)$, resulting in $\mathbf{B}_2(\psi) = \{[001, 111], [110, 111]\}$. Proceeding similarly, we obtain $\mathbf{B}_3(\psi) = \{[011, 111]\}$. Notice that intervals $X11$, $11X$, $1X1$ and 111 are discarded because they are contained in another intervals, while $0X1$ is discarded because it contains only don't cares.

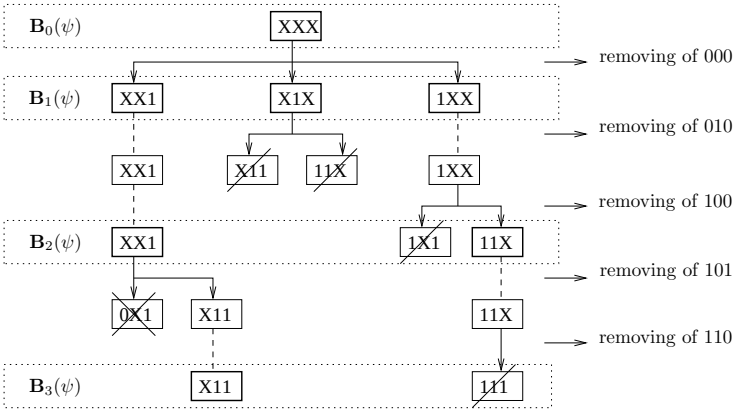


Figure 3. Example of the application of multi-level output ISI.

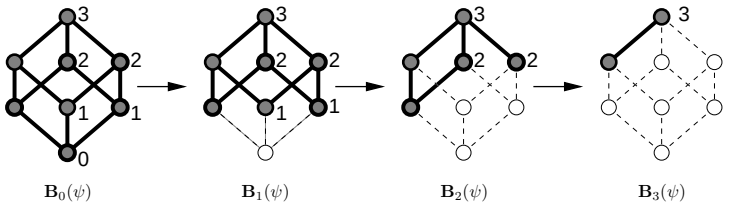


Figure 4. Lattice view of the intervals of Figure 3.

Given a partially defined ψ, let $\mathcal{M} = \{(X, l) : \psi(X) = l, l = 0, 1, \ldots, k\}$ (all elements whose label is known). The basis computation process is summarized in Algorithm 1.

4. Gray-level input and output

Gray-scale W-operators can be characterized by functions in the form $\psi : \{0, 1, \ldots, k_1\}^n \to \{0, 1, \ldots, k_2\}$, where k_1 and k_2 correspond to the input and output maximum gray-levels. Analogous to the previous case, the *kernel*

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

Algorithm 1 Basis computation.

$\mathbb{I} \leftarrow \{[\emptyset, W]\}$, with label 0; $\mathbb{I}_{\text{final}} \leftarrow \emptyset$;
$\mathcal{M}_1 = \mathcal{M}$;
// Repeat for each level, except k
for all $l \in \{0, 1, \ldots, k-1\}$ **do**
  $\mathcal{M}_0 = \{(X, i) \in \mathcal{M}_1 : i = l\}$; // elements to be removed
  $\mathcal{M}_1 = \mathcal{M}_1 \setminus \mathcal{M}_0$; // elements to be covered
  // Remove each $X \in \mathcal{M}_0$ from $\mathbb{I}$
  for all $(X, l) \in \mathcal{M}_0$ **do**
    $\mathbb{I}_{\text{New}} \leftarrow \emptyset$;
    $\mathbb{I}_{\text{P}} \leftarrow \{[A, B] \in \mathbb{I} : X \notin [A, B]\}$;
    $\mathbb{I}_{\text{tmp}} \leftarrow \{[A, B] \in \mathbb{I} : X \in [A, B]\}$;
    // Split each interval that contains X
    for all $[A, B] \in \mathbb{I}_{\text{tmp}}$ **do**
      $\mathbb{I}_{\text{split}} \leftarrow$ maximal intervals in $[A, B] \setminus \{X\}$;
      // Discard all intervals that does not cover any element in $\mathcal{M}_1$ and all those that are covered by another interval
      for all $[A', B'] \in \mathbb{I}_{\text{split}}$ **do**
        if $\exists X \in \mathcal{M}_1$ such that $X \in [A', B']$ **then**
          if $[A', B'] \not\subset [E, F], \forall [E, F] \in \mathbb{I}_{\text{P}}$ **then**
            $\mathbb{I}_{\text{New}} \leftarrow \mathbb{I}_{\text{New}} \cup \{[A', B']\}$;
          end if
        end if
      end for
    end for
    $\mathbb{I} \leftarrow \mathbb{I}_{\text{P}} \cup \mathbb{I}_{\text{New}}$;
  end for
  Label all intervals in $\mathbb{I}$ with $l + 1$;
  $\mathbb{I}_{\text{final}} = \mathbb{I}_{\text{final}} \cup \mathbb{I}$;
end for
Return $\mathbb{I}_{\text{final}}$;

of ψ at level i, $i \in \{0, 1, \ldots, k_2\}$, is the collection $\mathcal{K}_i(\psi)$ given by, for any $\mathbf{x} \in \{0, 1, \ldots, k_1\}^n$,

$$\mathbf{x} \in \mathcal{K}_i(\psi) \iff \psi(\mathbf{x}) \geq i, \tag{10}$$

and the *basis of ψ at level i*, $\mathbf{B}_i(\psi)$, is the collection of maximal intervals in $\mathcal{K}_i(\psi)$.

The sup-decomposition [3] of ψ in terms of kernel and basis are given, respectively, $\forall \mathbf{x} \in \{0, 1, \ldots, k_1\}^n$, by

$$\psi(\mathbf{x}) = max\{i \in \{0, 1, \ldots, k_2\} : \mathbf{x} \in \mathcal{K}_i(\psi)\}, \tag{11}$$

and

$$\psi(\mathbf{x}) = max\{i \in \{0, 1, \ldots, k_2\} : \mathbf{x} \in [\mathbf{a}, \mathbf{b}], [\mathbf{a}, \mathbf{b}] \in \mathbf{B}_i(\psi)\}. \tag{12}$$

In this section we present the extension of ISI for the computation of the basis. From the binary ISI we inherit the idea of splitting intervals, and from the multi-level output ISI we inherit the idea of a pyramid of basis.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

4.1 Algorithm

To simplify notations, we consider $k_1 = k_2 = k$ and $K = \{0, 1, \ldots, k\}$, and in order to generalize the splitting rule for this case, we introduce two auxiliary functions ξ_I e ξ_O. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be an element in K^W and let $O = (O_1, O_2, \ldots, O_n)$ and $I = (I_1, I_2, \ldots, I_n)$ be the smallest and the largest elements of K^W, respectively. We define

$$\xi_I(\mathbf{x}, i) = (I_1, \ldots, I_{i-1}, x_i - 1, I_{i+1}, \ldots, I_n) \tag{13}$$

and

$$\xi_O(\mathbf{x}, i) = (O_1, \ldots, O_{i-1}, x_i + 1, O_{i+1}, \ldots, O_n)\,. \tag{14}$$

Proposition 2. *Let* $[\mathbf{a}, \mathbf{b}]$ *and* $[\mathbf{c}, \mathbf{d}]$ *be two intervals in* K^W*, such that* $[\mathbf{c}, \mathbf{d}] \wedge [\mathbf{a}, \mathbf{b}] \neq \emptyset$*. Let* $\mathbf{a} = (a_1, a_2, \ldots, a_n)$*,* $\mathbf{b} = (b_1, b_2, \ldots, b_n)$*,* $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ *and* $\mathbf{d} = (d_1, d_2, \ldots, d_n)$*, where* $a_i, b_i, c_i, d_i \in K, \forall i \in [1, n]$*. The set of maximal intervals resulting from the subtraction of* $[\mathbf{c}, \mathbf{d}]$ *from* $[\mathbf{a}, \mathbf{b}]$ *is given by*

$$\big\{[\mathbf{a}, \mathbf{b} \wedge \xi_I(\mathbf{c}, i)] : i \in [1, n]\big\} \cup \big\{[\mathbf{a} \vee \xi_O(\mathbf{d}, i), \mathbf{b}] : i \in [1, n]\big\}. \tag{15}$$

Similarly to the binary case, when $\mathbf{c} = \mathbf{d} = \mathbf{x}$, where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, the above formula simplifies to

$$\big\{[\mathbf{a}, \mathbf{b} \wedge \xi_I(\mathbf{x}, i)] : i \in [1, n]\big\} \cup \big\{[\mathbf{a} \vee \xi_O(\mathbf{x}, i), \mathbf{b}] : i \in [1, n]\big\}. \tag{16}$$

After a removing operation, the number of resulting intervals as well as their dimension depend on the localization of the interval being removed. For an interval of dimension n, the number of resulting intervals may vary from n to $2n$. Figure 5 shows the three possibilities for the removing from an interval of dimension 2.

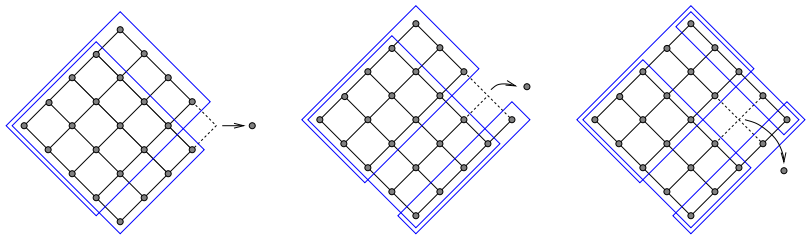


Figure 5. Three possible localizations of an element in an interval of dimension 2 and the intervals resulting from the splitting by that element.

Let $[A_1, B_1]$ and $[A_2, B_2]$ be two intervals such that there is a third interval $[C, D]$ that intercepts both (that is, $[C, D] \cap [A_1, B_1] \neq \emptyset$ and $[C, D] \cap [A_2, B_2] \neq \emptyset$). For the binary and multi-level output ISI, if we split both $[A_1, B_1]$ and $[A_2, B_2]$ by $[C, D]$, it is possible to show that no

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

interval resulting from the splitting of $[A_1, B_1]$ is contained in any interval resulting from the splitting of $[A_2, B_2]$, and vice-versa [8].

However, for the gray-level ISI, the same is not true. For example, in Figure 6, two intervals $[10, 44]$ and $[04, 44]$ are given, and the element to be removed is 24. From the splitting of $[10, 44]$ there results three intervals, $[30, 44]$, $[10, 43]$ and $[10, 14]$. From the splitting of the second interval there results $[04, 14]$ and $[34, 44]$. Analyzing the resulting intervals, we observe that $[34, 44] \leq [30, 44]$. This implies that the algorithm must, besides verifying whether the resulting intervals of a splitting by $[\mathbf{c}, \mathbf{d}]$ are not contained in intervals of $\mathbb{I}_{\mathrm{P}}$ (those that do not intercept $[\mathbf{c}, \mathbf{d}]$), also verify if they are not contained in an interval resulting from the splitting of another interval by $[\mathbf{c}, \mathbf{d}]$.

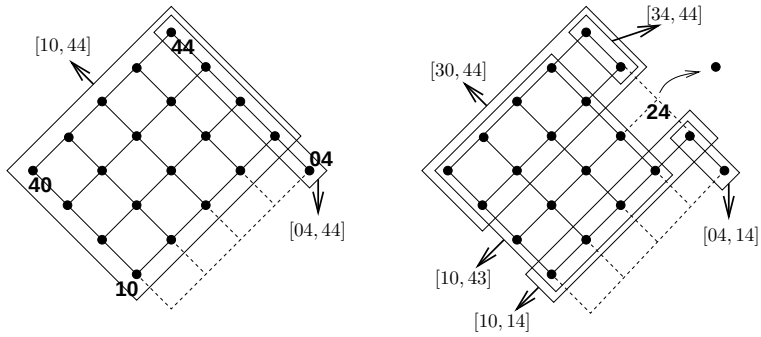


Figure 6. An interval resulting from the splitting of an interval may be contained in an interval resulting from the splitting of another interval by the same element.

As for the algorithm, the only modification needed with respect to the previously presented one is in Line 14, in order to include the verification discussed above. Of course, the splitting rule of Line 11 must be substituted by its equivalent for the gray-level case. Below follows Lines 14 to 16 of the modified algorithm, in order to compute the basis of a gray-level function.

if $[\mathbf{a}', \mathbf{b}'] \not\subset [\mathbf{e}, \mathbf{f}], \forall [\mathbf{e}, \mathbf{f}] \in \mathbb{I}_{\mathrm{P}} \cup \mathbb{I}_{\mathrm{New}}$ **then**
  $\mathbb{I}_{\mathrm{New}} \leftarrow \mathbb{I}_{\mathrm{New}} \cup \{[\mathbf{a}', \mathbf{b}']\}$;
end if

Figure 7 shows an example of the application of the gray-level ISI algorithm, step by step. The intervals obtained after removing each element are shown step by step in Figure 8. A minimum cover is computed each time all elements with a same label are removed.

Figures 8(b) to 8(e) show the intervals that result after removing elements with label 1, while Figures 8(f) to 8(h) show the intervals that result after removing elements with label 2. Intervals that do not cover any labeled elements are also removed and are not shown. Removing all elements with label 1 results in intervals that do not cover any element with label 1, but cover all elements of label greater than 1. Similarly, removing afterwards all

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

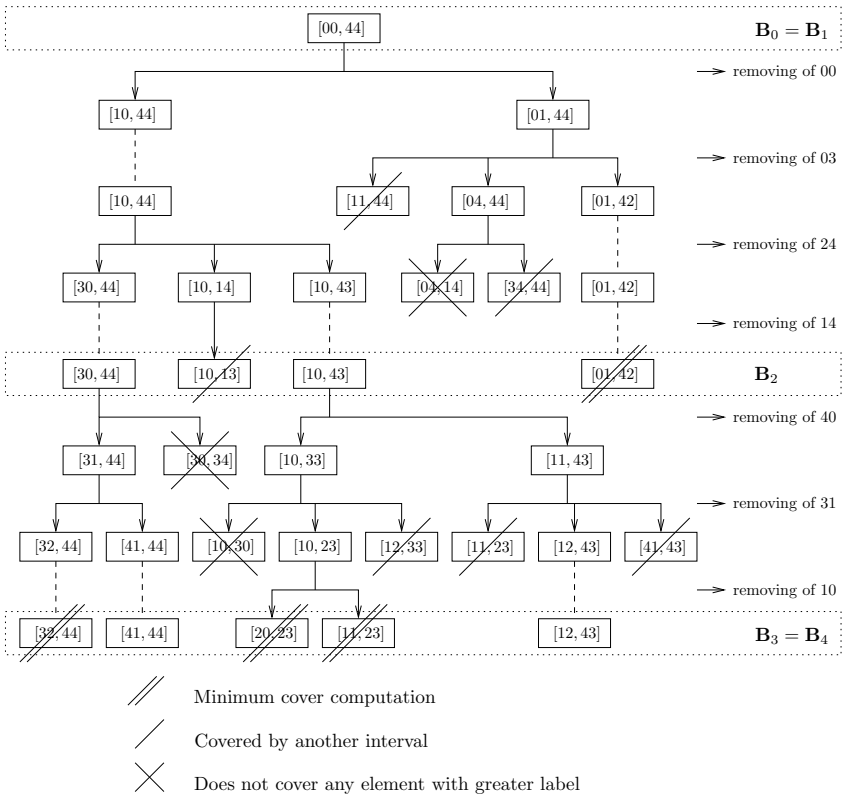


Figure 7. Example of the application of gray-level ISI.

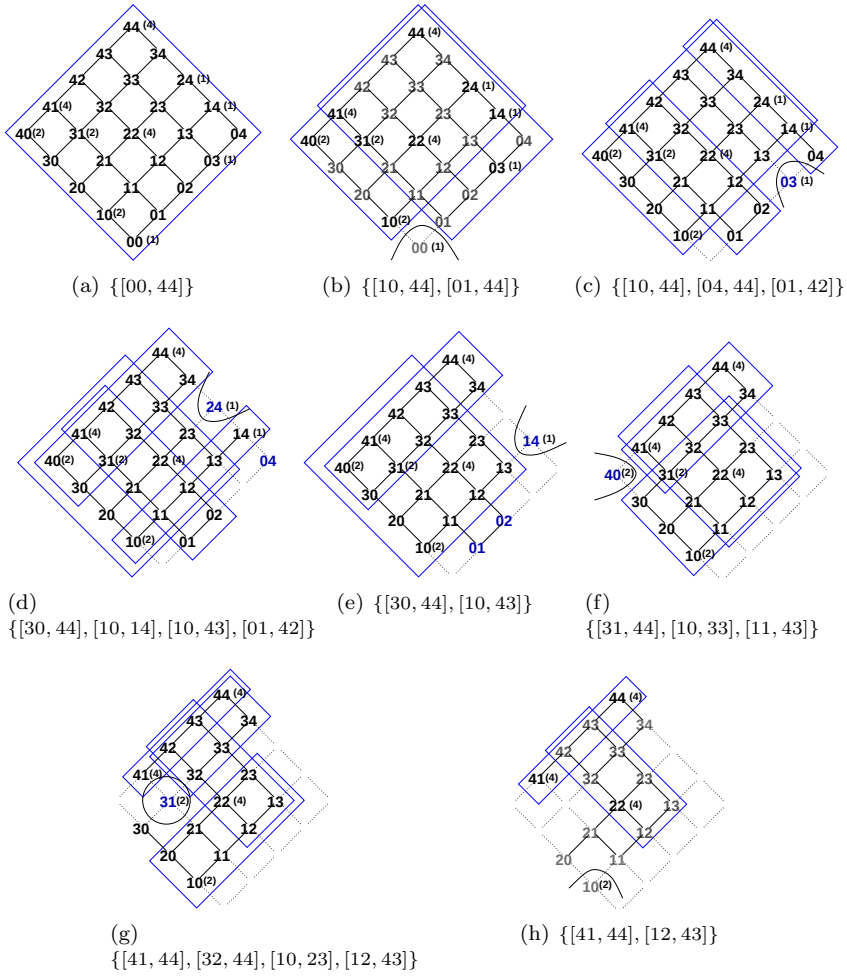
elements with label 2 results in intervals that cover all elements with label greater than 2, but do not cover any element with label 1 or 2.

The basis at level 0 ($\mathbf{B}_0$) is the initial interval and since there are no elements with label 0, it is also the basis at level 1 ($\mathbf{B}_1$). Four and three splitting operations, respectively, were performed to remove elements with label 1 from $\mathbf{B}_1$ and elements with label 2 from $\mathbf{B}_2$. Since there are no elements with label 3, the basis at level 4 is the same at level 3. The resulting basis is depicted in Figure 9.

5. Conclusion

We have reviewed and presented algorithms for the computation of maximal intervals contained in the kernel of binary and gray-level morphological operators. They are based on operations that remove non-kernel elements from intervals and express the resulting elements of the interval by means of a set of subintervals. Implementation of these algorithms must take into consideration the fact that kernels are not always entirely known. Our al-

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53



(a) {[00, 44]} (b) {[10, 44], [01, 44]} (c) {[10, 44], [04, 44], [01, 42]}

(d) {[30, 44], [10, 14], [10, 43], [01, 42]} (e) {[30, 44], [10, 43]} (f) {[31, 44], [10, 33], [11, 43]}

(g) {[41, 44], [32, 44], [10, 23], [12, 43]} (h) {[41, 44], [12, 43]}

Figure 8. Maximal intervals after removing elements with label 1 (removing of (a) 00, (b) 03, (c) 24, and (d) 14) followed by minimum cover computation, and after removing elements with label 2 (removing of (e) 40, (f) 31, and (g) 10) followed by minimum cover computation, for the example of Figure 8.

gorithms deal with cases in which some elements are known to be or not to be in the kernel, while nothing is known about the others. In the binary case, the algorithm corresponds to incompletely specified Boolean function minimization. Since these algorithms demand high computational cost, the use of smart data structures and good heuristics need to be considered. A simple example was given to illustrate the use of these algorithms as learning (or generalization) algorithms.

Further issues to be investigated include efficient implementation of the

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 15–26.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.20.01.53

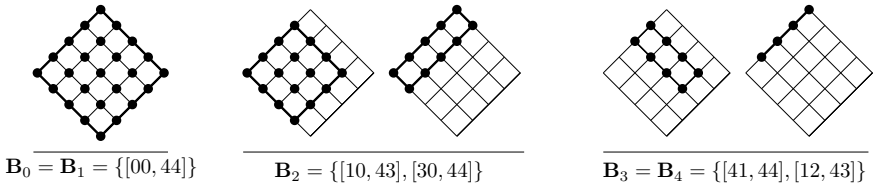


Figure 9. Resulting basis for the example depicted in Figure 8.

algorithm for the gray-scale case, conversion of an arbitrary representation of the kernel of image operators (such as decision trees, neural networks, support vector machines) to the basis representation, and efficient representation of the basis (aiming fast computation).

Acknowledgments

N. S. T. Hirata thanks FAPESP (Grant 04/11586-7) and CNPq (Grant 312482/2006-0). R. Hirata Jr. and J. Barrera thank CNPq.

References

[1] G. J. F. Banon and J. Barrera, *Minimal Representations for Translation-Invariant Set Mappings by Mathematical Morphology*, SIAM J. Applied Mathematics **51** (1991), no. 6, 1782-1798.

[2] ———, *Decomposition of Mappings between Complete Lattices by Mathematical Morphology, Part I. General Lattices*, Signal Processing **30** (1993), 299-327.

[3] J. Barrera, R. Terada, R. Hirata Jr, and N. S. T. Hirata, *Automatic Programming of Morphological Machines by PAC Learning*, Fundamenta Informaticae **41** (2000), no. 1-2, 229-258.

[4] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, *Logic Minimization Algorithms for VLSI Synthesis*, Kluwer Academic Publishers, 1984.

[5] H. J. A. M. Heijmans, *Morphological image operators*, Academic, Boston, 1994.

[6] F. J. Hill and G. R. Peterson, *Introduction to Switching Theory and Logical Design*, 3rd, John Wiley, 1981.

[7] N. S. T. Hirata, *Document Processing via Trained Morphological Operators*, To appear in Proceedings of ICDAR 2007, 2007.

[8] N. S. T. Hirata, J. Barrera, R. Terada, and E. R. Dougherty, *The Incremental Splitting of Intervals Algorithm for the Design of Binary Image Operators*, 6th International Symposium on Mathematical Morphology (H. Talbot and R. Beare, ed.), 2002, Proceedings of the 6th International Symposium: ISMM 2002, pp. 219-228.

[9] P. Maragos, *A Unified Theory of Translation-invariant Systems with Applications to Morphological Analysis and Coding of Images*, School of Elect. Eng. - Georgia Inst. Tech., 1985.

[10] ———, *A Representation Theory for Morphological Image and Signal Processing*, IEEE Transaction on Pattern Analysis and Machine Intelligence **11** (1989), no. 6, 586-599.

[11] G. Matheron, *Random Sets and Integral Geometry*, John Wiley, 1975.