DIMACS Technical Report 94-46
September 1994

Computing the Medial Axis Transform in Parallel with $O(1)$ scan operations

by

Afonso Ferreira[1,2]
CNRS - LIP
ENS Lyon – 69364
Lyon Cédex 07 France

Stéphane Ubéda[3]
TSI
Université Jean-Monnet
Saint Etienne France

[1]Visiting Member, partially supported by Dimacs NSF grant STC–91–199999 and the NJ Commission on Science and Technology.
[2]Partially supported by the PRC PRS and Math-Info of the French CNRS.
[3]Partially supported by the TIPI-TSI Project.

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

The main result of this paper shows that the *block-based digital medial axis transform* can be computed in parallel by a constant number of calls to *scan* (parallel prefix) operations. This gives time and/or work-optimal parallel algorithms for the *distance based* and the *block based* medial axis transform in a wide variety of parallel architectures. The most important contribution, however, is the practical aspects of our algorithms. In order to design such an algorithm, we first demonstrate that computing a block-based medial axis transform of a binary image reduces to computing the distance based medial axis transform of an oversampling of the image, because their labelings are equivalent.

1 Introduction

One of the most important problems in pattern recognition lies in shape representation, which is both an issue related to data reduction and data description. In [5], a shape descriptor named *medial axis* was described that allows a full reconstruction of the original shape and that has good properties for data reduction [13, 14].

The medial axis of a continuous object can be defined with the help of the *fire grass concept*, where the object is seen as a meadow, and a fire is lit along its contour such that all fire fronts invade the object with the same speed. The medial axis of the object is then the set of points reached by more than one fire front at the same time (see Figure 1-(I)). It can be also defined as the set constituted by the center of the maximal disks that recover the object (see Figure 1-(II)).

The medial axis is a powerful shape descriptor, whose most important properties include: an object and its medial axis are homotopic; the medial axis of an object is made of thick arcs and curves; and finally, if any point of the medial axis of an object is labeled with the radius of its corresponding maximal disk, then the object can be exactly rebuilt using the information of its medial axis [17].

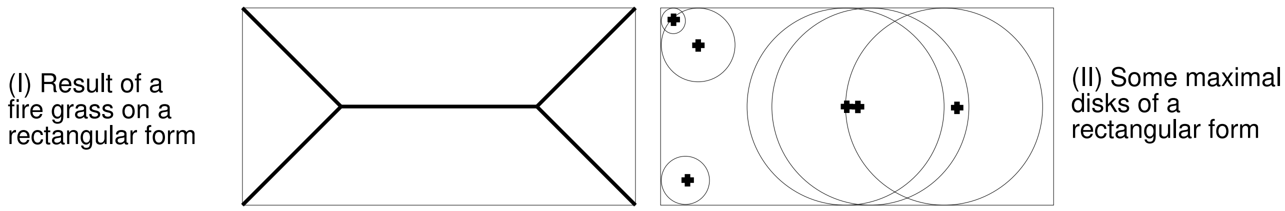


Figure 1: Illustration of the continuous medial axis

The subject of this paper is the *digital medial axis* of an object from a binary image, which is a derivative of the medial axis originally proposed in [5]. Unfortunately, it is well known that such transpositions, from continuous to digital cases, raise many problems [6, 18]. In this case, working in a discrete space implies that fire wavefronts are symbolized by contour lines, and disks become of squared form. Therefore, the digital Medial Axis Transform (MAT for short) is defined as a *recovering of the object by maximal digital disks included in the object*, where a digital disk is said to be *maximal* if it is not included in any other digital disk contained in the object [15].

In this paper, we focus on the parallel processing of two types of MAT, namely, the distance-based MAT (denoted DB-MAT) which corresponds to a set of pixels included in the object, each labeled with its distance to the background of the picture and the block-based MAT (denoted BB-MAT) which is based on the notion of recovering the object by maximal square blocks of pixels of any parity size. We consider images as $N \times N$ square sets of points of a digital grid E, each point being associated to a binary value, 0 to denote a background pixel, and 1 to denote an object pixel.

1.1 Previous work

As seen above, the serial medial axis, as well as its utilisation in image processing and shape recognition, have been extensively studied. Besides the papers referenced thus far, of great importance for our results is the computation of the medial axis on the *oversampled* image that was described in [3]. Most of these results will be recalled in the next Section.

In the parallel setting, also, extensive research has been carried out in connection to the computation of the medial axis transform of binary images. The distance based extraction of a medial axis transform is related to the distance transform of the image. Therefore, this latter problem received considerable attention in the literature. In particular, Bitz and Kung proposed a systolic implementation in [4]. Based on such algorithm, Miguet and Robert designed a distance transform algorithm for MIMD computers ([11]). Regarding SIMD algorithms, Lu and Varman studied the distance transform on mesh-connected computers in [10], and Jenq and Sahni obtained a parallel algorithm for a dynamic programming solution of the medial axis transform – block-based version –, that can be implemented on a CREW PRAM with N^2 processors, in time $O(\log N)$ [8]. Finally, a work-optimal EREW-PRAM algorithm was given by Kim in [9], in time $O(\log N)$ with $N^2/\log N$ processors.

1.2 This paper

The work-optimal algorithm from [9] is based on a divide-and-conquer approach where the original problem of size S is partitioned into problems of size $\sqrt{S}$, and this recursively. Therefore, it is of complex implementation even in the EREW-PRAM. For more practical architectures as Hypercube multiprocessors or real parallel systems, such an algorithm would be very difficult to program and would have a larger complexity.

The main result presented in this paper is as follows:

> *A parallel algorithm to compute the block-based MAT of an object of an $N \times N$ binary image, that consists only of a constant number of calls to scan operations.*

Our algorithm is the first that is practical and yields time and/or work-optimal implementations of DB-MAT in various parallel architectures. For instance, in contrast with [9], it can be work-optimally implemented also in Hypercubes in time $O(\log N)$ with $N^2/\log N$ processors.

Let MAT(t,p) denote that MAT can be solved in parallel time t with p processors in a given model, and $t_s(S)$ denote the time required to perform the scan operation over S elements in a given parallel machine. Our results imply that:

- In real parallel systems, we have MAT($O(t_s(N^2) + N^2/p)$,p).
- In the Hypercube, we have MAT($O(\log N)$,$N^2/\log N$);
- In the EREW PRAM, we have MAT($O(\log N)$,$N^2/\log N$);

In order to derive our algorithm, we first demonstrate the following property of binary images, which is general enough to be interesting in its own right in the field of image processing.

> *The labeling of a distance-based MAT of an oversampling of a binary image yields a labeling of a block-based MAT of the original image.*

Then, computing BB-MAT of a binary image reduces to computing DB-MAT of the oversampled image, which is much easier. Our approach is very attractive in the sense that the algorithm is quite simple, all the complexity being left to its proofs of correctness.

Our paper is organized as follows. In Section 2 we survey the main types of medial axis transforms, recalling definitions, properties, and algorithms. Then, in Section 3 we prove that the labels of a block-based medial axis transform of a binary image can be computed from the labels obtained by a distance-based medial axis transform of the oversampled image. With this fact, we are able to design a new sequential algorithm for BB-MAT – which has the same time complexities as those in the literature –, that is less sequential in nature. Finally, its scan-based parallelization is introduced in Section 4, where its time and processors complexities are proven. We close the paper with some concluding remarks and directions for further research.

2 On discrete medial axis transforms

In this Section we recall some definitions and sequential algorithms for the different forms of MAT.

2.1 Grids, distances, disks, and blocks

Let $E \subset \mathcal{R}^2$ be a grid generated by vectors $\vec{k_1}$ and $\vec{k_2}$ as follows:

Definition 1 $E = \{x \in \mathcal{R}^2 \mid \exists (i,j) \in \mathcal{Z}^2,\ x = i\vec{k_1} + j\vec{k_2}\}$

Over E, the d_4 distance (or *city-block*) and the d_8 distance (or *chessboard*) are defined as:

Definition 2 *Let $x = (x_1, y_1)$ and $y = (x_2, y_2)$ be two points of E.*

1. *The d_4 distance between x and y is $d_4(x,y) = |x_1 - x_2| + |y_1 - y_2|$.*
2. *The d_8 distance between x and y is $d_8(x,y) = max(|x_1 - x_2|, |y_1 - y_2|)$.*

Using these two distances, two corresponding forms of neighborhood, 4-neighborhood (or V_4) and 8-neighborhood (or V_8) can be defined:

Definition 3 *Let $x \in E$. Then,*

1. $V_4(x) = \{y \in E \mid d_4(x,y) = 1\}$,

2. $V_8(x) = \{y \in E \mid d_8(x,y) = 1\}$.

The notion of a r radius digital disk depends on the distance and the neighborhood, and is defined as:

Definition 4 *Let x be a pixel in E, a d_4-disk of radius r centered in x is defined as*

$$D_4(x,r) = \bigcup_y \{y \in E \mid d_4(x,y) \leq n\}$$

$$D_8(x,r) = \bigcup_y \{y \in E \mid d_8(x,y) \leq n\}.$$

A d_8-disk of radius r can be seen as a square of $(2r-1) \times (2r-1)$ object pixels. Hence, a d_8-disk is a square with odd size. The definition of *block* generalizes this notion.

Definition 5 *A k-block in an object is a group of $k \times k$ pixels included in the object.*

Note that a k-block may be well represented by a label k associated to its upper-left corner pixel.

In [2], considering the grid E which is the support of the image, a derived grid of E was defined.

Definition 6 *The derived grid E' of a grid E is defined as:*

$$E' = \{x' \in R^2 \mid \exists (i',j') \in Z^2,\ x = i'\vec{k'_1} + j'\vec{k'_2}\}$$

with

$$\vec{k'_1} = \frac{\vec{k_1} + \vec{k_2}}{2} \ \ and\ \ \vec{k'_2} = \frac{\vec{k_2} - \vec{k_1}}{2}.$$

The derived grid E' is composed by the pixels in the original grid E and the pixels in a dual grid E^+ obtained by a 45^o rotation of E, such that $E^+ = E' \setminus E$. In this way, E' will be the support grid for the *oversampling* points to be computed for the original image in E. Furthermore, in the derived grid E' we shall use the notions of distance (d'_4 and d'_8) and neighborhood (V'_4 and V'_8), that are the extensions of the usual notions, under the new directions $\vec{k'_1}$ and $\vec{k'_2}$.

2.2 A distance based MAT algorithm

Below, we use distance-based MAT (denoted DB-MAT) only for the two distances previously defined (d_4 and d_8). It is clear, however, that it is possible to define a DB-MAT using more complex distance definitions, such as the Chanfrein distance, for instance [1].

For d_4 and d_8, extraction of the corresponding DB-MAT has been characterized using the Distance Transform of the image [16]. The Distance Transform of an object from an $N \times N$ binary picture in E consists of labeling each pixel i of the object with the size of the maximal disk centered in i that is included in the object. Again, one can use the Distance Transform of an image according to d_4 or to d_8 (see Figure 3). Since the Distance Transform of a binary object measures the distance of each pixel to the background, its computation can be done in $O(N^2)$ operations, using two raster-scanning of the whole image, one forward and one backward. See Figure 2 for an algorithm computing the Distance Transform for d_8. Based on the Distance Transform, a distance-based MAT can be defined as follows.

```
begin
  for i := 1 to N
    for j := 1 to N
      if I(i,j) ≠ 0 then
        I(i,j) := 1 + min(I(i−1,j−1), I(i−1,j), I(i−1,j+1), I(i,j−1))
    endfor
  endfor
  for i := N to 1
    for j := N to 1
      if I(i,j) ≠ 0 then
        I(i,j) := min(I(i,j), I(i+1,j+1)+1, I(i+1,j)+1, I(i+1,j−1)+1, I(i,j+1)+1)
    endfor
  endfor
end
```

Figure 2: Sequential Distance Transform.

Definition 7 *Let $S \in E$ be an object. The distance-based MAT in d_4 of S is such that:*

$$d_4\text{-}DB\text{-}MAT(S) = \{p \in S \mid \forall q \in S,\ D_4(p, r_p) \not\subset D_4(q, r_q)\},$$

where

$$D_4(p, r_p) \text{ and } D_4(q, r_q) \text{ are maximal } d_4 \text{ disks in } S$$

The d_8-DB-MAT is defined analogously. Therefore the centers of disks of the d_4-DB-MAT (or d_8-DB-MAT) correspond to local maxima in the Distance Transform of the image, with their corresponding radius equal to the label of each center [12]. A pixel p labeled r_p in the Distance Transform is part of the DB-MAT if, for all pixel in its neighborhood, there is

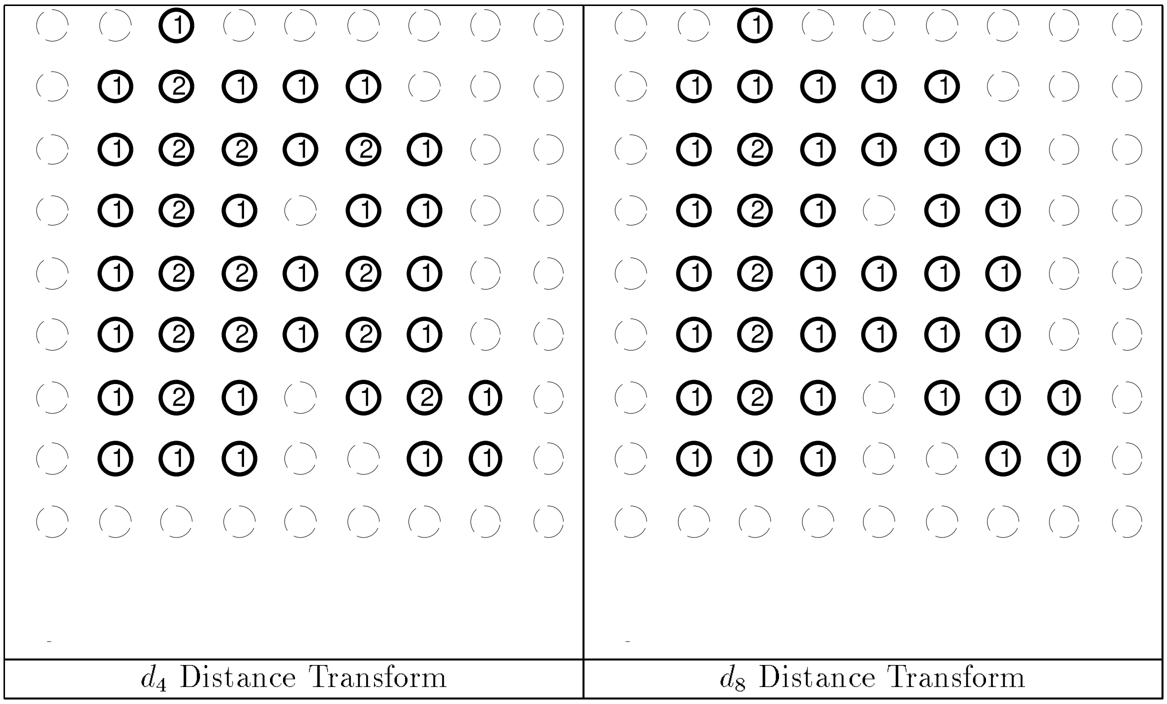


Figure 3: Distance Transform of a binary picture.

no pixel with a label in the Distance Transform that is larger then r_p. Figure 4 shows the resulting labels of a d_4-DB-MAT and a d_8-DB-MAT in a binary image. Therefore, DB-MAT can be also characterized as in the definition 8, below.

Definition 8 *A pixel $x \in E$ is a pixel of the DB-MAT in a given metric, if it is a locally non null maximum with regard to its Distance Transform value in the same metric.*

As shown in [2, 3], a DB-MAT can also be computed in the derived grid, having some interesting properties, the most important being the thickness of the obtained medial axis transform. In the following we will show how to use this kind of DB-MAT in order to design a BB-MAT of the original image.

2.3 A block-based MAT algorithm

MAT was defined as a recovering of the object S by maximal disks that are contained in S. Unfortunately, the notion of "disk" would imply that in a d_8-DB-MAT objects are covered by squares of odd-size length only. However, we recall that the notion of "disk" can be replaced by any desired standard shape, so, we consider blocks rather than disks. A BB-MAT consists therefore of a recovering of an object S by maximal squares with sides of any parity length (i.e., blocks of pixels).

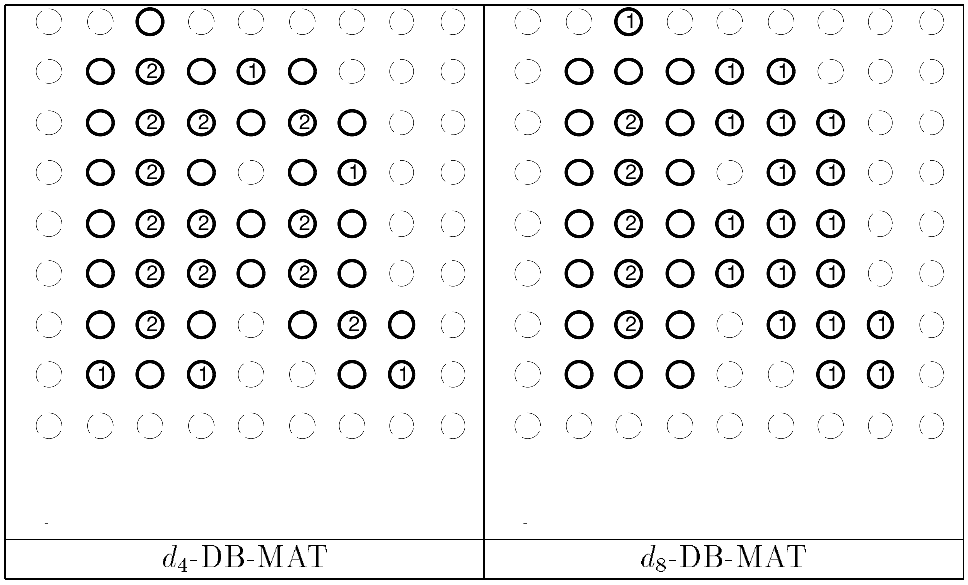


Figure 4: Distance-based digital Medial Axis Transform.

Another remark is that there would be a problem of representation for such a BB-MAT, since the center of a block is not always a pixel in E. Thus, we shall represent a block in E by its upper-left corner, labeled with the length of its sides. The algorithm presented in Figure 5 computes a BB-MAT in $O(N^2)$ sequential time, with this kind of representation [8]. Figure 6 shows the result of this algorithm. We recall that finding the minimum number of rectangles that cover a given object S is NP-hard [7].

In the following Section we introduce some relationships among MATs in order to derive a new BB-MAT algorithm.

3 A new BB-MAT algorithm

In the following, we shall show how to use the DB-MAT in the derived grid in order to compute the BB-MAT in the original grid. For this purpose, we prove that the labeling obtained for the first can be used to compute the labels for the second.

3.1 The relationship among MATs

In the previous Section we saw the definition of the dual grid E' and of the set of new points created by such a transformation E^+. The pixels in E^+, since they represent the oversampling of the image in E, are not labeled as yet. Actually, we can label a pixel $x \in E^+$ as object (label 1) or background (label 0) according to some specific rule, as follows.

```
begin
  for k := 1 to N do
    i := N − 1; j := N − k
    while j < N do
      if I(i, j) = 0 then M(i, j) = 0
      else
        M(i, j) =min(M(i, j + 1), M(i + 1, j), M(i + 1, j + 1)) + 1
      endif
      i := i − 1; j := j + 1
    endwhile
  endfor
  for k := N + 1 to 2N − 1 do
    j := 0; i := 2N − 1 − k
    while j ≥ 0 do
      if I(i, j) = 0 then M(i, j) = 0
      else
        M(i, j) =min(M(i, j + 1), M(i + 1, j), M(i + 1, j + 1)) + 1
      endif
      i := i − 1; j := j + 1
    endwhile
  endfor
end
```

Figure 5: Sequential Block-Based Medial Axis Transform algorithm.

- either x is labeled object if *all* its 4-neighbors in V'_4 belong to the object (a morphological dilatation [17]). Notice that all such neighbors belong to E.

- or x is labeled object if there is *at least one* pixel labeled object in the V'_4 neighborhood of x (a morphological opening (i.e., a dilatation followed by an erosion) [17]).

Concerning an object S, each labeling rule above defines a different oversampling of S.

Definition 9 *Let $S \subset E$ be an object. The oversamplings of S, $D_1(S)$ and $D_2(S)$, are as follows:*

$$D_1(S) = S \cup \{x' \in E^+ \ : \ |\{V'_4(x') \cap S| = 4\},$$

and

$$D_2(S) = S \cup \{x' \in E^+ \ : \ \exists x' \in S, \ x \in V'_4(x)\}.$$

Before studying MAT algorithms in the derived grid, we give some structural results on the connectivities of S and $D_i(S)$, $i = 1, 2$, that we borrow from [3].

Proposition 1 *An object S is 4-connected in E if and only if $D_1(S)$ is 8-connected in E'. An object S is 8-connected in E if and only if $D_2(S)$ is 4-connected in E'.*

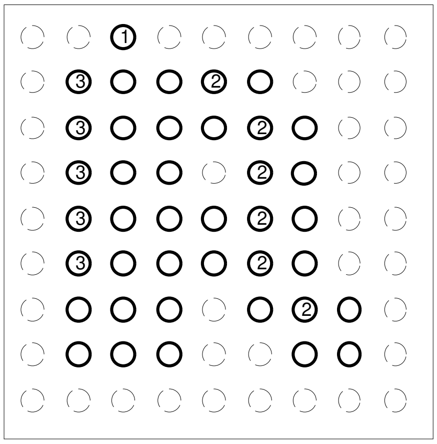


Figure 6: Result of the BB-MAT computation.

In Figure 7 one can see some objects in the grid E and their corresponding D_1 oversamplings. It is easy to see that a L-size block in E yields a d'_4-disk of radius $(L-1)$ in E' as its corresponding D_1 oversampling. Notice, however, that not all 4-connected sets in E' can be obtained as a D_1 oversampling of an object in E.

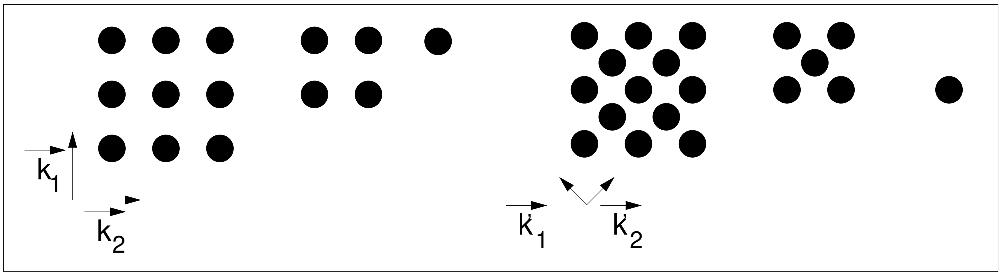


Figure 7: D_1 applied to some L-side blocks in E.

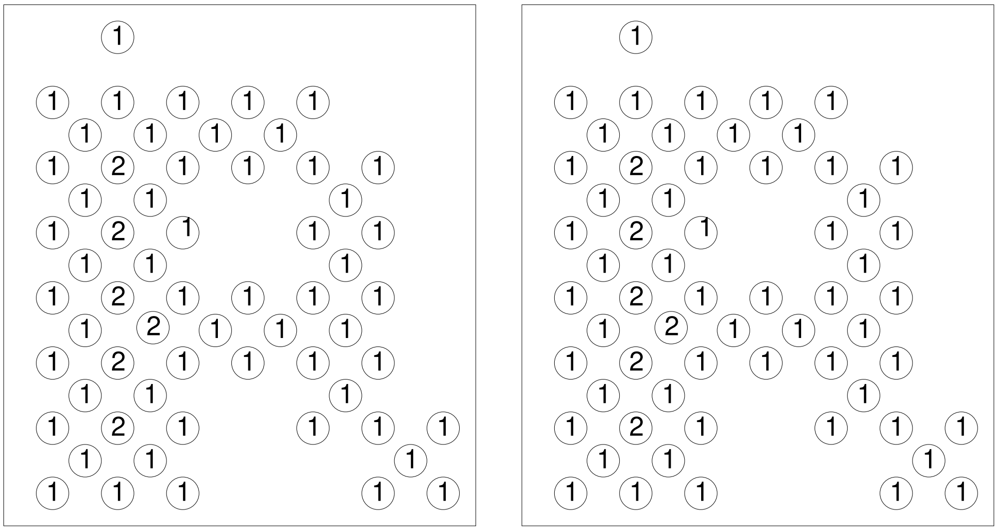
3.2 A new algorithm for BB-MAT

Now we introduce a new way of computing the block-based Medial Axis Transform. The idea is to first build the oversampling of the object S, namely $D_1(S)$. Then we compute DB-MAT of $D_1(S)$ in E', that happens to be easily transformed in the BB-MAT of S in E, as we are going to show.

For this, we need to define a kind of reverse operator for D_1, which shall be denoted D_1^{-1}.

Definition 10 *The set $D_1^{-1}(S')$, where $S' \subset E'$, is the set of pixels S in E such that $D_1(S) = S'$, if it exists.*

Note that such an operator is not defined for all sets $S' \subset E'$. On the other hand, it is not difficult to see that D_1^{-1} is defined for any d'_4-disk centered in E^+ with an odd radius and



Based on D_1 Based on D_2

Figure 8: Derived Medial Axis Transform.

also for any d_4'-disk centered in E with an even radius. Further, we can prove the following lemma.

Lemma 1 *Let S' be an object in E' obtained by a D_1 oversampling. Then, for any maximal d_4'-disk δ in S' it holds that:*

a) *if it is centered in E^+ then it has an odd radius;*

b) *it it is centered in E then it has an even radius.*

Proof. In the following we prove statement (a), the proof of (b) being analogous.

Suppose that δ has an even radius r. Then, the pixels in the border of δ belong to E^+ and, by definition, are labeled 1. Therefore, there is a d_4'-disk centered in E' whose radius is larger than r, what is a contradiction since δ is maximal. $\square$

Therefore,

Corollary 1 *The operator D_1^{-1} is defined for any maximal d_4'-disk centered in E^+ and also for any maximal d_4'-disk centered in E.*

Finally, we can state the main results of this Section. The operator D_1^{-1} was originally defined over a set of pixels. We informally extend this notion such that this operator is also defined over any set Λ composed of sets of pixels. Again, we say that $D_1^{-1}(\Lambda)$ is defined only if D_1^{-1} is defined for all sets of pixels in Λ.

Theorem 1 *Let S' be an object in E' obtained by a D_1 oversampling of an object S in E. Then Δ is a set of maximal d'_4-disks that recover S' if and only if the set $D_1^{-1}(\Delta)$ is a set of maximal blocks that recover S.*

Proof. Let Q be the set of all blocks in S (remark that $D_1^{-1}(\Delta) \subseteq Q$), and R be the set of all disks in S'.

($\Rightarrow$) Let Δ be the set of maximal d'_4-disks that recover S' and suppose that $D_1^{-1}(\Delta)$ is not composed only of maximal blocks of S. Then,

$$\exists \beta \in D_1^{-1}(\Delta), \text{ and } \exists \beta' \in Q, \text{ such that } \beta \subset \beta',$$
$$\Rightarrow D_1(\beta) \subset D_1(\beta'), \text{ by definition},$$
$$\Rightarrow D_1(\beta) \not\subset \Delta,$$
$$\Rightarrow D_1^{-1}(D_1(\beta)) \not\subset D_1^{-1}(\Delta),$$
$$\Rightarrow \beta \not\subset D_1^{-1}(\Delta). \text{ Contradiction.}$$

($\Leftarrow$) Let $D_1^{-1}(\Delta)$ be a set of maximal blocks of S and suppose that $D_1(\Gamma)$ is not the set of maximal d'_4-disks of S'.

Then,

$$\exists \delta \in \Delta, \text{ and } \exists \delta' \in R, \delta' \text{ maximal, such that } \delta \subset \delta',$$
$$\Rightarrow D_1^{-1}(\delta) \subset D_1^{-1}(\delta'), \text{ by definition},$$
$$\Rightarrow D_1^{-1}(\delta) \not\subset D_1^{-1}(\Delta),$$
$$\Rightarrow D_1(D_1^{-1}(\delta)) \not\subset D_1(D_1^{-1}(\Delta)),$$
$$\Rightarrow \delta \not\subset \Delta. \text{ Contradiction.}$$

□

Since the $D_1^{-1}(S)$ transformation conserves all points of E, we have

Corollary 2 *Let an object S in E be given. Then, the set of blocks represented by D_1^{-1}(d'_4-DB-MAT($D_1(S)$)) is a locally maximum block recovering of S.*

And the folowing result give us an algorithmic way to find the BB-MAT of an object in E from a DB-MAT of the oversampled image in E'.

Corollary 3 *Let an object S in a grid E be given, as well as the d'_4-DB-MAT of $D_1(S)$ in E'. Further, let l_i be the label associated to pixel $i = (x', y')$ of $D_1(S)$ by the computation of d'_4-DB-MAT. Then, $(l_i - 1)/2$, associated to pixel $j = (x', y' + l_i)$, $i = 1 \dots N^2$, represent the labels of a BB-MAT of S in E.*

Proof. It is not difficult to see this fact, if we recall that the disk of radius l_i computed by the d'_4-DB-MAT in E' is a block of size $(l_i - 1)/2$ in E, and that the upper pixel of the disk in E' is the left upper corner of the block in E. □

Hence, computing the BB-MAT of a binary image is reduced to computing the d'_4-DB-MAT of the oversampled image in E'.

4 Optimal parallel algorithms for BB-MAT

In this Section, we shall show how to use the results developed in the previous Sections in order to propose a parallel algorithm for BB-MAT that is composed only of a few calls to scan procedures.

As shown in Section 2, Distance Based MAT in the derived grid E' can be computed using the Distance Transform of the oversampled image. Then, by corollary 3, the BB-MAT of the original image can be computed from the d'_4-DB-MAT of the oversampled image. In the following we study the complexity of such an algorithm.

Let us focus on the parallel computation of the Distance Transform. Recall that the Distance Transform of a binary image is a labeling of the binary image, where each pixel of the object has a label with the distance of the shortest path to the background. We note d_4-DT (respectively, d_8-DT) the Distance Transform under d_4 (respectively, d_8).

The parallel algorithm for Distance Transform is made of calls to a procedure of the type "Segmented Scan". Let us consider an array of binary values (0 or 1). A *right oriented* segmented scan (RightSegmentedScan) computes, for each 1 valued element in the array, the shortest distance to a 0 valued element to the left of the array, the result of the scan being 0 for 0 valued elements. A *left oriented* segmented scan (LeftSegmentedScan) can be defined analogously. If we take the minimum value between the result of a left and right oriented segmented scans for each 1 valued element of a binary array, we compute its shortest path to a 0 valued element. Thus, the algorithm shown in Figure 9 computes the d_4-DT in parallel. Therefore, we proved that.

begin
  $H_1(i,j)$:= LeftSegmentedRank(M,i)
  $H_2(i,j)$:= RightSegmentedRank(M,i)
  $H(i,j) := \min(H_1(i,j), H_2(i,j))$
  $V_1(i,j)$:= TopSegmentedRank(M,j)
  $V_2(i,j)$:= BottomtSegmentedRank(M,j)
  $V(i,j) := \min(V_1(i,j), V_2(i,j))$
  $V'(i,j) := V(i,j) - H_1(i,j)$
  $MATH_1(i,j)$:= LeftSegmentedMin(V', i)+$H_1(i,j)$
  $V'(i,j) := V(i,j) - H_2(i,j)$
  $MATH_2(i,j)$:= RightSegmentedMin(V', i)+$H_2(i,j)$
  $H'(i,j) := H(i,j) - V_1(i,j)$
  $MATV_1(i,j)$:= TopSegmentedMin(H', i)+$V_1(i,j)$
  $H'(i,j)$:= H(i,j)−$V_2(i,j)$
  $MATV_2(i,j)$:= BottomSegmentedMin(H', i)+$V_2(i,j)$
  $MAT(i,j)$:= Min$(MATH_1, MATH_2, MATV_1, MATV_2)$
end

Figure 9: Parallel algorithm for the d_4-DB-MAT.

Theorem 2 *Let a binary image of size $N \times N$ be given. Then, the d_4-DT can be implemented in parallel by a constant number of scan operations.*

If we add to the previous algorithm the same kind of scan computations over the diagonals, then we have the following.

Theorem 3 *Let a binary image of size $N \times N$ be given. Then, the d_8-DT can be implemented in parallel by a constant number of scan operations.*

Computing the DB-MAT from the Distance Transform is not difficult. We must keep only those pixels which are maximal according to their labels, i.e., those whose Distance Transform is at least as large as the one of all the neighbors. Therefore, once we have the Distance Transform of a binary image, only local computations are required to compute the DB-MAT of the image, yielding that

Theorem 4 *Let a binary image of size $N \times N$ be given. Then, the d_4-DB-MAT and the d_8-DB-MAT can be implemented in parallel by a constant number of scan operations.*

Finally, to compute the BB-MAT of an object S, we start by building the $D_1(S)$ oversampling of the object S. Then, we compute the DB-MAT of $D_1(S)$ in E', and we use this result to build the BB-MAT of S in E, according to corollary 3. Hence, by theorems 2, 3 and 4, and corollary 3, we have

Theorem 5 *Let a binary image of size $N \times N$ be given. Then, the block based digital medial axis transform can be computed in parallel with a constant number of scan operations.*

Corollary 4 *Let a binary image of size $N \times N$ be given. Then, the block based digital medial axis transform can be computed in parallel with the following time and processor complexities, depending on the architecture chosen.*

a) *Real parallel systems: $O(t_s + N^2/p)$ time with p processors, where t_s is the time required to perform the scan operation over N^2 elements;*

b) *Hypercube multiprocessors: $O(\log N)$ time with $N^2/\log N$ processors;*

c) *EREW-PRAM: $O(\log N)$ time with $N^2/\log N$ processors.*

5 Conclusion

We used the derived grid of a binary image in order to design a new parallel algorithm for the block based medial axis transform, that is composed only of a constant number of calls to scan operations. The use of scan operations makes the algorithm time and/or work-optimal in various parallel architectures, and further exemplifies the power of the scan operator.

In fact, once the theoretical results on the relationship between the derived and the original image have been proved, the algorithm itself is quite easy to be programmed, making it very attractive for implementation in real parallel systems.

As directions for further research, it would be very interesting to study the impact of the results proved in Section 3 on the design of other parallel algorithms for binary image processing.

References

[1] C. Arcelli and G. Sanniti di Baja. Finding local maximal in pseudo-euclidean distance transform. *Computer Vision, Graphics and Image Processing*, (43):361–367, 1988.

[2] G. Bertrand. Détermination de l'axe médian par sur-échantillonnage fictif. In $4^{\grave{e}me}$ *congrés RFIA, Afcet-Inria (Paris)*, pages 265–275, 1984.

[3] G. Bertrand. Skeleton in derived grids. In 7^{th} *ICPR (Montreal)*, pages 326–329, 1984.

[4] Bitz and Kung. Path planning on the warp computer: Using a linear systolic array in dynamic programming. *International Journal of Computer Mathematics*, (25):173–188, 1988.

[5] H. Blum. *A Transformation For Extracting New Descriptors.* Symp. On Models for perception of speech and visual form, MIT Press, 1964.

[6] J.M. Chassery and A. Montanvert. *Géométrie discrète en analyse d'images.* Editions Hermés, Paris, 1991.

[7] M.R. Garey and D.S. Johnson. *Computers and Intractability.* San Francisco, W.H. Freeman, 1979.

[8] J.F Jenq and S. Sahni. Serial and parallel algorithms for medial axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(12):1218–1224, 1992.

[9] S. K. Kim. Optimal parallel algorithms for region labeling and medial axis transform of binary images. *Siam Journal of Discrete Mathematics*, 4(3):385–396, 1991.

[10] M. Lu and P. Varman. Optimal algorithms for rectangle problems on a mesh-connected computer. *J. Parallel Distributed Comput.*, 5:611–622, 1988.

[11] S. Miguet and Y. Robert. Path planning on a ring of processors. *International Journal of Computer Mathematics*, (32):61–74, 1990.

[12] A. Montanvert. Medial line: Graph representation and shape description. In *Eighth International Conference on Pattern Recognition (Paris, 1986)*, pages 430–432, 1986.

[13] T. Pavlidis. A review of algorithms for shape analysis. *Comp. Graph. and Image Proc.*, 7:243–258, 1978.

[14] A. Rosenfeld and A. C. Kak. *Digital Picture Processing.* Academic Press, New-York, 1989.

[15] A. Rosenfeld and J.L. Pfaltz. Sequential operations in digital pictures. *Journal of the ACM*, 13(4):69–72, 1966.

[16] A. Rosenfeld and J.L. Pfaltz. Computer representation of planar regions by their skeletons. *Communication of the ACM*, (10):119–125, 1967.

[17] J. Serra. *Image Analysis and Mathematical Morphology.* Academy Press, 1982.

[18] A.Y. Wu, S.K. Bhaskar, and A. Rosenfeld. Computation of geometric properties from the medial axis transform in $o(nlogn)$ time. *Computer Vision, Graphics and Image Processing*, 34:76–92, 1986.