Θ (1) Time Parallel Algorithm for Determining

the Diameters of Multi-Level Image on

the Reconfigurable Mesh Computer Machine

Mohammed Khaldoun

NEST Research Group, LRI Laboratory E.N.S.E.M, Hassan II University
Casablanca/Morocco

Jelloul Elmesbahí

NEST Research Group, LRI Laboratory E.N.S.E.M, Hassan II University
Casablanca/Morocco

Soukaina Ihirri

NEST Research Group, LRI Laboratory E.N.S.E.M, Hassan II University
Casablanca/Morocco

Ahmed Errami

NEST Research Group, LRI Laboratory E.N.S.E.M, Hassan II University
Casablanca/Morocco

Copyright © 2017 Mohammed Khaldoun et al. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

The recent advances in research on physics, computing, medicine and others, require the use of certain shape characterization parameters, especially for morphological analysis. The most known morphometric characteristics are: the perimeter, the diameter, the circularity, the tortuosity, the compactness factor and others. In this paper we propose a new parallel algorithm for the diameter determination of multi-level image in Θ(1) time, intended for a parallel Reconfigurable Mesh Computer (RMC) machine of n x n Elementary Processors.
Mohammed Khaldoun et al.

The algorithm consists in determining the maximum diameter of each component of a 2D multilevel image. The approach used involves only the points of the contour of each component of the image. This allowed us to apply this approach on a multi-level image. Using the $\Theta(1)$ time Min / Max algorithm, we have been able to reduce the complexity of our algorithm to $\Theta(1)$ time.

Keywords: Parallel image processing, Reconfigurable Connected Mesh, diameter

1. Introduction

The diameter of objects in an image represents today an indispensable tool for the characterization and classification of forms. It is used as a measurement tool in a multitude of disciplines such as medical imagery[13], agriculture[1] [8], geology [7] [9] [15], nanotechnology [14], Composite Materials, Microstructures [10] [5], Microbiology and Medicine Biology [2] [18]. Most of the algorithms used in these fields are essentially based on determining the diameter of the smallest convex polygon that surrounds the component. This polygon can be either a rectangle, or in the most general case the convex envelope of the component. The diameter of a convex polygon $P$ is defined by the greatest distance which separates any pair of points $(p, q)$ from this polygon:

$$diam(P) = \max \{\text{dist}(p, q)\} \quad (1)$$

Initially this approach was proposed by Shamos in 1978 [4] to find the diameter of a convex polygon. In 1983, Toussaint [16] demonstrated that the same algorithm serves to solve other geometric problems with a complexity of $O(n)$. A fairly exhaustive summary of the work using this type of approach can be found in document [3]. Note that all these algorithms concern sequential architectures. In the case of parallel architectures, the work of Dyer and Rosenfeld [12] which explained how to determine the external diameter with a complexity of $O(n)$ on a machine type MCC (Mesh-Connected -Computer), composed of nxn PEs. But their method was based on the use of non-Euclidean distances. MILLER and STOUT [6] subsequently proposed a solution for determining the external diameter based on the calculation of the Euclidean distances with a complexity of $O(n)$ for the same type of machine. But all these works are limited to the case of bi-level images.

In this paper, we present a new parallel algorithm for the external diameter determination in RMC machine of n x n process elements. Our algorithm can calculate with a complexity of $\Theta(1)$ the diameter of all components in the case of multi-level image. This algorithm uses a new approach, based only on the Euclidean distances measured on the contour of the component. It does not necessitate the preliminary formation of the convex envelope of the component. This approach allowed reducing the complexity of our solution compared to those proposed in the literature.
The aim contribution of this paper is summarized as follows:

- Our proposed method guarantees $\Theta(1)$ time processing, because all the steps are executed in $\Theta(1)$ time.
- Our proposed method provides the diameter of not only the bi-level images with the presence of holes, but also for the multi level’s one.
- Our proposed algorithm is based on reducing the amount of information to be processed. The diameter in our method is provided by considering the contour of each component first, and then a group of 16 points of the contour pixels are in charge of computing the maximum distances beyond all PEs of the group in order to provide two points. These two points refer to the extreme points of an object.

In this paper, section 2 is devoted to the computational model used to the implementation of the proposed algorithm. Section 3 presents our new diameter extraction algorithm. Section 4 shows results of both bi-level and multi-level image diameter extraction. This paper concludes with some remarks on the future works.

2. Concerned model

The computational model of machine used in this paper is a parallel Reconfigurable Mesh Computer (RMC) [17] of size nxn Elementary Processors (PEs) arranged in a square matrix. It is a machine which respects the SIMD (Single instruction multiple data) model, in which each PE is located in the matrix by its row and column coordinates and can be characterized by its identifier ID = n.i + j, where i and j represent the row number and the column number respectively. In this architecture, Figure 1 (a) refers to a set of PEs, each one is connected to its four neighbors; if they exist; through communication channels. It has a finite number of registers, of bus length $\log_2(n)$ bits, in which it stores data to perform arithmetic and logic operations. All PEs can perform reconfiguration operations to exchange information with other PEs in the mesh. Figure 1. (b) describes the set of possible configurations of each PE.

![Figure 1: (a) Concerned model, (b) Possible configurations of the PE](image-url)
The Communication of each PE with its neighbors is implemented through four different operations:

(i) **Single Bridge (SB)**

A PE of the RMC is in the SB state, if it establishes links between two of its communication channels, either in transmitting mode or the receiving one. Also, it may disconnect some of its communication channels. Figure 1.b) shows the six possible configurations (1-6) of the SB configurations.

(ii) **Double Bridge (DB)**

A PE is in DB state, when it achieves configuration involving two independent buses. The configurations from 7 to 10 of FIG 1.b) shows the three possible cases of DB.

(iii) **Cross Bridge (CB)**

A PE goes to the CB state if it connects all its active communication channels into one. The configuration 11 of FIG 1. b) Is the only possibility of the CB operation.

(iv) **Direct Broadcast**

The direct broadcast operation consists in transmitting information from a transmitting PE to a set of receiving ones. The procedure of implementing this operation is achieved as follows:

1. All PEs are in CB mode. The receiving PE are coupled by their receiving ports. While the transmitting PEs are in the transmitting mode via their ports.
2. The transmitting PE sends the information throughout its ports. Then, all other PE receive the same information.

### 3. Proposed algorithm

#### 3.1 Description of the proposed algorithm

The aim of this section is to describe the new diameter determination approach. We assume that our algorithm is able to determine the diameter of 2-D images in \( \Theta (1) \) time, using the RMC architecture. The proposed method is based on Min/Max algorithm. This algorithm; as described in paper [11]; explains the method by which determines in \( \Theta (1) \) time the PE of the contour that has the minimum (or maximum) ID among all the other PEs of the contour. In our case, we applied this algorithm to determine the PE of the contour that is at a minimum (or maximum) Euclidean distance from a reference point.

#### 3.2 Elementary step

Before turning to describe the proposed algorithm, a preliminary Contour isolation procedure has to be implemented in order to reduce the amount of information to be processed. It is performed in constant time (\( \Theta (1) \)) operations. This procedure isolates all the PEs on the edge of the component. It is carried out as follows:

(i) Each PE compares its gray level with its four neighbors.
(ii) Each PE that has at least one different gray level is marked.
(iii) Each marked PE blocks its communication channels with PEs having a gray level different than its value.

From now throughout, we make the following assumptions, which are applied depending on the state of the PE in the concerned sub steps of the algorithm:

- Each computational PE may be configured as a receiver of a transmitter after the execution of any sub step of the algorithm.
- Each PE may execute the $\Theta(1)$ Min/Max procedure.
- Each PE of the contour may computes the Euclidean distance.

### 3.3 Diameter determination

The execution of our algorithm for determination of maximum diameter of each component of images requires four steps, all of which are executed in constant time ($\Theta(1)$ time):

- **Step 1:** The determination of the quadrilateral that includes the component to be treated (Figure 2).
- **Step 2:** The determination of the extreme points of the component which are closest to the four vertices of the quadrilateral. These points form the first subgroup of points: $SG_1$ (Figure 3).
- **Step 3:** Search for the points of contour that belong to each of the four sides of the quadrilateral. These points form the second subgroup of points: $SG_2$ (Figure 7).
- **Step 4:** Search for the maximum diameter of the component (Figure 8).

Now, we will present a detailed description of each of these four steps:

(i) **Step 1:** The aim of this step is to determine the quadrilateral which includes the component. By determining $X_m = x_{min}$, $X_M = x_{max}$, $Y_m = y_{min}$ and $Y_M = y_{max}$. Automatically, the coordinates of the vertices $S_1$, $S_2$, $S_3$ and $S_4$ of the quadrilateral will be known (Figure 2). The vertices are used in order to form the group of points; belonging to the contour; that participate in the diameter determination.

![Figure 2: the coordinates of the vertices S1, S2, S3 and S4 of the quadrilateral](image-url)
The four pairs of coordinates of the quadrilateral $S_1 (X_m, Y_m)$, $S_2 (X_M, Y_m)$, $S_3 (X_M, Y_M)$ and $S_4 (X_m, Y_M)$ are determined by the following eight sub steps:

1. All the PEs of the contour execute the MIN algorithm for their abscissa ($X$) in order to find the PE with the minimum abscissa.

2. The PE of the contour having the minimal abscissa $X_m$ diffuses this value to all the PEs of contour which have been configured as receivers after the MIN algorithm.

3. All contours PE execute the MAX algorithm using their abscissa ($X$).

4. The PE of the contour with the maximum abscissa $X_M$ diffuses this value to all the PEs of the contour which have been configured as receivers after the MAX algorithm.

5. All the PEs of the contour execute the MIN operation for their ordinates ($Y$).

6. The PE with the minimum ordinate $Y_m$ diffuses this value to all the PEs of the contour which have been configured as receivers after the MIN algorithm.

7. All PEs belonging to the contour execute the MAX operation for their $Y$-coordinates.

8. The PE of the contour with the maximum ordinate $Y_M$ diffuses this value to all the PEs of the contour which have been configured as receivers after the MAX algorithm.

After these eight sub steps, all the PEs of the contour of a given gray level component, know the values $X_m$, $X_M$, $Y_m$ and $Y_M$, therefore the vertices $S_1 (X_m, Y_m)$, $S_2 (X_M, Y_m)$, $S_3 (X_M, Y_M)$ and $S_4 (X_m, Y_M)$.

(ii) **Step 2**: In this step, we determine the set of PEs $\{R_{11}, R_{12}, R_{21}, R_{22}, R_{31}, R_{32}, R_{41}, R_{42}\}$ which belong to the contour and closest to the 4 vertices of the quadrilateral. Figure 3 illustrates the position of these points. These PEs, constitute the first subgroup $SG_1 = \{R_{11}, R_{12}, R_{21}, R_{22}, R_{31}, R_{32}, R_{41}, R_{42}\}$. This subgroup takes part in the diameter extraction.
\( \Theta(1) \) time parallel algorithm for determining the diameters …

Figure 3: the set of PEs belonging to the contour and closest to the 4 vertices of the quadrilateral.

The determination of the elements of SG1 is based on minimal Euclidean distance calculation between each vertex taken as reference (\( S_i \in \{ S_1, S_2, S_3, S_4 \} \) ) and all the PEs of the contour. This procedure is executed in a constant time (\( \Theta(1) \) time). In Figure 4, the dark marked PE represents the vertex S2, while the PEs marked in clear (A, B, C, D, E); belong to the nearest portion of the contour of the concerned component.

Figure 4: the nearest portion of the contour to the vertex S2.

This procedure of finding the set of elements is carried out in three successive sub steps:

1. The PE\( j \) of the contour defined by (\( X_j, Y_j \)) as coordinates are computing the Euclidean distances \( d_{jS2} \) separating them from the vertex S2 with coordinates (\( X_{S2}, Y_{S2} \)).

\[
    d_{jS2} = \sqrt{(X_j - X_{S2})^2 + (Y_j - Y_{S2})^2}
\]  

(2)
2. These PE\textsubscript{j}s execute the Min algorithm to identify the PE which has the minimum distance : $\text{Min} (d_{jS2})$

3. All PE\textsubscript{j}s retain the coordinates of the PE which has the minimum distance. This distance separates this PE to the vertex S2 of the quadrilateral. This PE will be considered as one of the points of the subgroup $\text{SG1}= \{R11, R12, R21, R22, R31, R32, R41, R42\}$

We mention that the result of this third sub step may produce a single candidate point in some cases (Figure 5 (a)) or several candidate points (Figure 5 (b)) equidistant to the reference located on arc of circle of center S2. In the case of one candidate, this point is located at a minimum distance from the reference S2. While, in the case of several points, the two extreme points of the arc of a circle; containing these candidate points; must be marked (Figure 5 (c)).

In order to find the two extreme PE (PE\textsubscript{1} and PE\textsubscript{2}) of the arc of a circle, the sub steps mentioned before must be completed with the three additional sub steps follows:

4. All the contour’s PEs with the Min distance are marked.

5. All the PEs belonging to the contour; except those that have been marked; go into simple bridge operation SB (See Figure 8 (d)) in order to exchange data beyond the marked ones.

6. Each marked PE detects whether it is between two marked PEs:
   - If yes, it is configured in SB
   - If not, it will be one of the extreme PEs with minimal Euclidean distance to S2: as it is seen in Figure 5 (e). In the case where phase 3 releases a single candidate point, it will be duplicated in the subgroup $\text{SG1}$. Example $R11 = R12$ of the Figure 6.

We mentioned that the same sub steps described for the vertex S2 are then executed for each of the three other vertices S1, S3 and S4 of the quadrilateral. After the execution of all sub steps of step 2 according to all vertices, all marked PE; identified by R11, R12, R21, R22, R31, R32, R41 and R42; send their coordinates to all PE belonging to the contour in order to save this coordinates for a subsequent use in the diameter determination process.
$\Theta (1)$ time parallel algorithm for determining the diameters ...

Figure 5: The different case of candidates with the minimum distance that separate the PE to the vertex S2. (a): single candidate. (b): several candidates on arc of circle. (c): the two extreme points on arc are marked. (d): all PE on the arc are marked. (e): marked PE between two marked PE are configured in SB.

Figure 6: Case of duplicated point in the sub group SG
(iii) **Step 3:** In this step, we determine the candidate points belonging to the 4 sides of the quadrilateral. These PEs constitute the second subgroup of extreme PEs: \( \text{SG2} = \{E, F, G, H, I, J, K, L\} \) (see Figure. 7).

The position of these PEs is chosen depending on the four references relative to the four sides of the quadrilateral: \(X_m (\text{xmin}), X_M (\text{xmax}), Y_m (\text{Ymin})\) and \(Y_M (\text{ymax})\).

The procedure of finding the extreme PEs belonging to the sides of the quadrilateral is executed in 8 successive sub steps, that allow to identify the PEs \(E, F, G, H, I, J, K, L\) of the contour component. In each sub step, we applied the Min/Max algorithm [18] to the values of the abscissa and ordinates relating to the quadrilateral. The set of PEs belonging to this sub group are determined as follow:

1. Determining the PE whose ordinate coincides with the ymin (Ym) axis of the quadrilateral, and which has a minimal abscissa.
2. Determining the PE whose ordinate coincides with the axis ymin (Ym) of the quadrilateral, and which has a maximum abscissa.
3. Determining the PE whose ordinate coincides with the axis ymax (YM) of the quadrilateral, and which has a minimal abscissa.
4. Determining the PE whose ordinate coincides with the ymax (YM) axis of the quadrilateral, and which has a maximum abscissa.
5. Determining the PE whose abscissa coincides with the axis Xmin (Xm) of the quadrilateral, and which has a minimal ordinate.
6. Determining the PE whose abscissa coincides with the axis Xmin (Xm) of the quadrilateral, and which has a maximum ordinate.
7. Determining the PE whose abscissa coincides with the axis Xmax (XM) of the quadrilateral, and which has a minimum ordinate.
8. Determining the PE whose ordinate coincides with the axis Xmax (XM) of the quadrilateral, and which has a maximum ordinate.

After the execution of all the sub steps of step 3, the PEs that have been marked as \(E, F, G, H, I, J, K, L\) respectively, send their coordinates to all other PEs around the component. As a result, they all know the coordinates of the PEs that forms the subgroup: \(\text{SG2} = \{E, F, G, H, I, J, K, L\}\).

In order to compute the diameter of each component, a group of PEs is formed by the union of the two sub groups \(\text{SG1} \) and \(\text{SG2}\) obtained after the execution of the steps 2 and 3 of the proposed algorithm. This group \(G\) contains 16 candidates in charge of computing the diameters.

\[G = \text{SG1} \cup \text{SG2} = \{R11, R12, R21, R22, R31, R32, R41, R42, E, F, G, H, I, J, K, L\}.\]
(iv) \textbf{Step 4:} In this step, we determine the Max diameter of the component. This step is carried out in two successive sub steps:

1. Each PE of the Group G, calculates the Euclidean distances $d_{nm}$ which separates it from the others of the same Group:
   \[ d_{nm} = \sqrt{(X_n - X_m)^2 + (Y_n - Y_m)^2} \text{ with } n, m \in \{1, \ldots, 16\} \text{ et } n \neq m \]

2. Each PE$_n$ calculates the maximum value of the Euclidean distances $d_n = \text{Max} \ (d_{nm})$ (see figure 8) and retains the coordinates $(X_m, Y_m)$ of the PE corresponding to this maximum distance.

The execution of these first two sub steps by each PE of the group G makes it possible to obtain a set D formed of 16 distances: $D = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}, d_{13}, d_{14}, d_{15}, d_{16}\}$. Thus, a PE$_n$ of group G knows the distance $d_n$ of the group D associated with it and the coordinates $(X_m, Y_m)$ of the PE$_m$ which corresponds to this distance.

3. All the PE$_j$ of group G execute the algorithm Max [18] to identify the two PEs which have the maximum distance and represent the extreme points of the object, where $D_{\text{max}} = \text{Max} \ \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}, d_{12}, d_{13}, d_{14}, d_{15}, d_{16}\})$. The two PEs \{PE$_n$, PE$_m$\} of the group G, refer to the two points that define the maximum diameter, thus represent the two extreme points of the component. These two PEs execute the following sub step 4 to identify the master who is in charge of broadcasting the max diameter through the contour. While the other PEs of the group G go into the mode SB.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure7.png}
\caption{the 16 points in charge of computing the diameter}
\end{figure}
Mohammed Khaldoun et al.

Figure 8: (a): The set of 16 distances of the Group G. (b): represent the maximum distance separating R22 and R42

4. PE\textsubscript{n} and PE\textsubscript{m} execute the Min (or Max) algorithm [18] to identify the one with the minimum or maximum ID. The PE with the minimum ID for example, transmits the two coordinates \((X_n, Y_n)\) and \((X_m, Y_m)\) to all the other PEs of the contour. Thus, all the PEs of the contour of the component know the coordinates of the PEs which are the ends of the line segment forming the diameter. In the example of FIG. 8 (b), the two extreme PEs are marked respectively by R22 and R42. And the maximum distance corresponds to the distance \(d_8\)

### 4. Experimental results

We have scrupulously translated all the steps that make up this parallel algorithm into a serial algorithm. Through used the MATLAB simulation, we verify the effectiveness of the proposed algorithm, and we obtained satisfactory results. We give in the figures 9, some examples of obtained results:

- (a), (b), (c), (d) represent image of animals with two levels of gray: chicken, elephant, fish and ant.
- (e), (f) represent two levels of gray images with holes. Examples are crowned and cup of coffee.
- (g), (h) represent images with multi gray level. We were able to extract the diameter of each area of the map and also the diameter of all flowers in a Carpet.
$O(1)$ time parallel algorithm for determining the diameters …

Figure 9: simulation results. (a), (b),(c) and (d): illustrate the diameter extraction in the case of bi-level images.(e),(f): diameter extraction in the case of bi-level images with the presence of holes. (g): the diameter extraction in the case of a map. (h): the external diameter of all flowers in a Carpet.
From the obtained results, one can see that we were able to determine the diameter of not only the bi-level image with the presence of holes, but also when it’s about multi-level images. That is one of the strengths of our proposed method especially when the result are obtained in a constant number of operations (Θ(1) time).

5. Conclusion

We have presented in this paper a new parallel algorithm for an RMC machine to calculate the diameters of the related components that form a multi-level image. The approach that has been used involves only the points of the contour of each component of the image. This allowed us to apply it on a multi-level image. Using the algorithm that determines the Min / Max in Θ (1) time, we have been able to reduce the complexity of our algorithm to Θ(1) time.

In our future work, we aim to exploit this new approach to determine other characteristics of related components while keeping the complexity Θ (1) time.

References


(1) time parallel algorithm for determining the diameters of the minimal spanning tree (MST),

\[O(1)\] time parallel algorithm for determining the diameters...


Received: March 16, 2017; Published: April 21, 2017