Global Journal of Computer Science and Technology: F
Graphics & Vision
Volume 21 Issue 1 Version 1.0 Year 2021
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

ϴ (1) Time Parallel Algorithm for Finding 2D Convex Hull on a Reconfigurable Mesh Computer Architecture

By Jelloul Elmesbahi, Mohammed Khaldoun, Omar Bouattane & Ahmed Errami

Hassan II University

Abstract- In this paper we propose a parallel algorithm in image processing in ϴ(1) time, intended for a parallel machine " Reconfigurable Mesh Computer (RMC), of size n x n Elementary Processors (PE). The algorithm consists in determining the convex envelope of a two-level 2D image with a complexity in ϴ(1) time. The approach used is purely geometric. It is based solely on the projection of the coordinates of PEs retained in specific quadrants and on the application of the algorithm that determines the Min / Max in ϴ (1) time.

This has reduced the complexity of the algorithm for determining the convex hull at ϴ (1) time.

Keywords: processional image, parallel processing, reconfigurable mesh computer, convex hull.

GJCST-F Classification: G.1.0

ϴ (1)Time Parallel Algorithm for Finding 2D Convex Hull on a Reconfigurable Mesh Computer Architecture

Jelloul Elmesbahi [α], Mohammed Khaldoun [σ], Omar Bouattane [ρ] & Ahmed Errami [Ѡ]

Abstract- In this paper we propose a parallel algorithm in image processing in ϴ(1) time, intended for a parallel machine " Reconfigurable Mesh Computer (RMC), of size n x n Elementary Processors (PE). The algorithm consists in determining the convex envelope of a two-level 2D image with a complexity in ϴ(1) time. The approach used is purely geometric. It is based solely on the projection of the coordinates of PEs retained in specific quadrants and on the application of the algorithm that determines the Min / Max in ϴ (1) time.

This has reduced the complexity of the algorithm for determining the convex hull at ϴ (1) time.

Keywords: processional image, parallel processing, reconfigurable mesh computer, convex hull.

I. Introduction

The problem of the convex envelope (convex hull) of a plane shape has been the subject of several studies in recent years. This very important problem concerns several areas such as image processing, pattern recognition or robotics [FU97]. The convex envelope can be used, for example, to normalize a shape, to triangulate a set of points, to extract topological characteristics, to decompose a complex shape in order to facilitate its recognition.

Compared to the very important number of works that have been realized on the use of sequential algorithms to extract the convex hull, the number of parallel algorithms to solve this problem is significantly lower. We have to wait until the beginning of the 80's to see the first parallel algorithms, which calculate the convex hull of a set of points [CHO81], [NAT81], [ALD83].Since then, several authors have presented parallel algorithms to solve this problem with different levels of complexity {[KIM87], [LIN93], [PRA89] and on different input data as for example on a set of points in a plane or in a space, on bi-level images, on multi-level images with a single component or several components. On the other hand, the algorithms proposed in these works, were designed for different models of machines such as the model Mesh Connected Computer [MIL85], the model Polymorphic Torus [LI89] or the model Reconfigurable Mesh Computer [HAY98].

The synthesis of these different works leads us to note that the diversity of the parameters put into play complicates any attempt at global comparison between the performances of the different approaches. Table 1 presents a synthesis of some bibliographical references on this problem by specifying each time the machine model used, the nature of the input data as well as the degree of complexity obtained.

Among the parallel algorithms for the determination of the convex hull of the connected components of a multi-level image, mention may be made of [ERR05], which describes an approach based on the structural characterization algorithm. As for the algorithm [BOU02] applies a purely geometric method. Our algorithm calculates with a complexity of ϴ (1) the convex hull of an image or set of points with two levels of gray. This algorithm uses a purely geometric approach. It is based solely on the projection of the coordinates of PEs retained in specific quadrants and on the application of the algorithm that determines the Min / Max in ϴ (1) time [ELMES91]. This allowed us to reduce the complexity of our solution compared to those proposed in the literature. The contribution of this document is summarized as follows:

- The proposed method guarantees constant time processing because all steps are executed in ϴ (1) time, while most existing algorithms are based on image processing multiple times.
- The proposed provides the convex hull not only for a set of points but also for images with two gray levels.

Author α σ Ѡ: NEST Research Group, LRI Laboratory E.N.S.E.M, Hassan II University, Casablanca, Morocco.
e-mails: j.elmesbahi@ensem.ac, aerrami@yahoo.fr, m.khaldoun@ensem.ac.ma
Author ρ: ENSET Mohammedia, Hassan II University, Morocco etc.
e-mail: o.bouattane@gmail.com

Table 1: Synthesis of some works on the problem of the extraction of the convex hull

Machine model	Nature of data	Complexity	Bibliography
Mesh connected Computer	Points Sets	θ(log n)	[MIL85]
Polymorphic Torus	Points Sets	O(1) par ensemble de points	[LI89]
hypercube	Points Sets	θ(log n) O(log n)	[MIL88] [STO88]
Pyramid	Points Sets	O(log n)	[STO88]
Pyramid tree Mesh of tree	Points Sets	θ($\log^3 n/(\log \log n)^2$)	[MIL88]
Reconfigurable Mesh	Points Sets	θ(log n)	[MIL88]
EREW PRAM	Points Sets	θ(log n)	[MIL88]
Reconfigurable Mesh	Sorted Points Sets	O(log log n)^2	[HAY98]
Reconfigurable Mesh	Bi-level image one component	O($\log^2 n$) by component	[MIL93]
Reconfigurable Mesh Computer (RMC)	Multilevel image several components	O (log n)n is the number of segments that approximate the contour of the largest component of the image.	[BOU02]
Reconfigurable Mesh Computer (RMC)	Multilevel image several components	O (log n)n represents the total number of extreme points present on the contour of the largest component of the image	[ERR05]

In this article, section 2 is devoted to the calculation model used for the implementation of the proposed algorithm. Section 3 presents the algorithm for determining the convex hull. Section 4 presents the result of determination of the convex hull of a two-level image matrix. This paper concludes with some remarks on future work.

II. Computational Model Architecture

The computational model of machine used in this paper is a parallel Reconfigurable Mesh Computer (RMC) [ELMES86] 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. Figure1. (b) describes the set of possible configurations of each PEThe communication of each PE with its neighbors is implemented through four different operations:

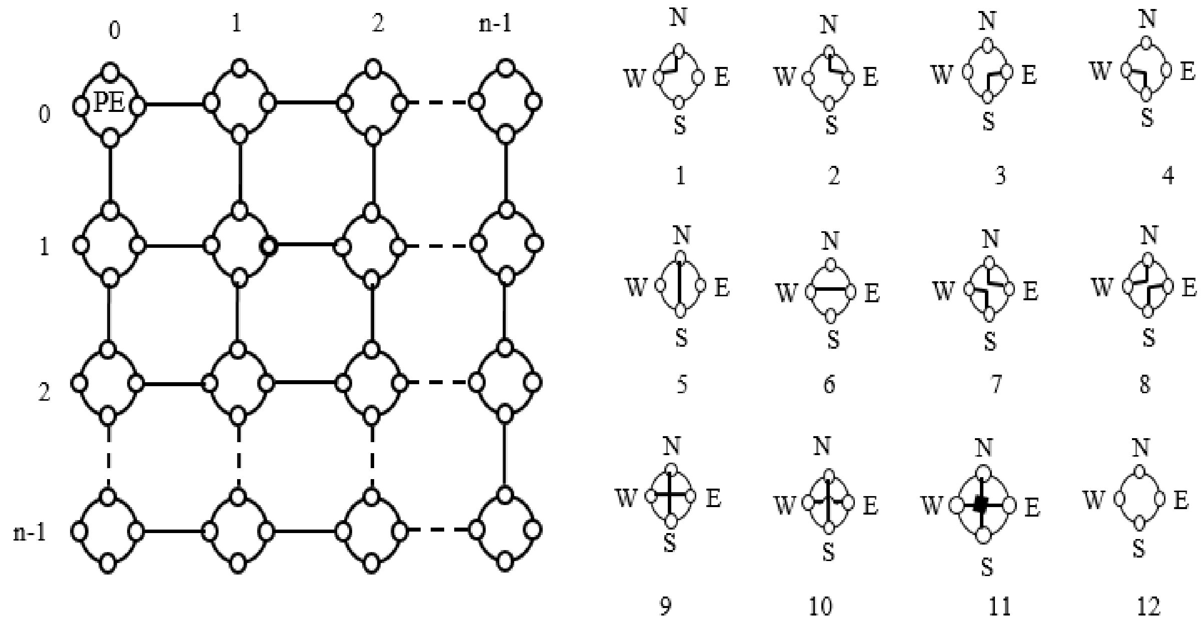
Single Bridge (SB): A PE of the RMC is in the SB state, if it establishes links between two of its communication channels, either in transmitting mode or the receiving one. In addition, it may disconnect some of its communication channels. Figure 1.b) shows the six possible configurations (1-6) of the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

Double Bridge (DB): A PE is in DB state, when it achieves configuration involving two independents buses. The configurations from 7 to 10 of Figure 1.b) shows the three possible cases of DB

Cross Bridge (CB): A PE goes to the CB state if it connects all its active communication channels into one. The figure 1 (b) shows the only possible configuration (11) with the CB operation.

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

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

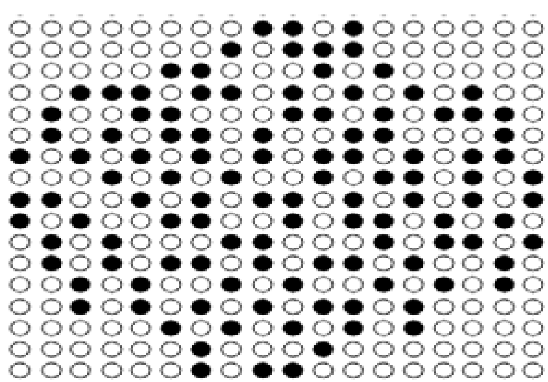


(a) (b)

Figure 1: (a) Concerned model, (b) possible configurations

III. θ (1) Time Algorithm to Determine the Convex Hull of a Bi-level Image

Either to determine the convex hull of a set of points or PEs (Elementary Processors) arranged in the form of an image matrix as shown in Figure 2.



●Marked PE○Unmarked PE

Figure 2: Elementary Processors (PEs) arranged in the form of a set of points representing points of an image matrix.

The algorithm for determining the convex hull consists of 7 steps:

1. *Step 1:* Determination of points: i_m, i_M, j_m, j_M and points A, B, C, D, E, F, G and H (Figure 3).
2. *Step2:* Determination of points: I_1', I_2', I_3' et I_4' : (Figure 10).
3. Step 3: Determination of points: I_1 , I_2, I_3 et I_4 : (Figure 3).
4. *Step 4:* The delimitation of the area P_1 (quadrant: NW), then the determination of the PEs belonging to the convex hull forming part of this zone. The procedure will be applied to zones P_2, P_3 and P_4. (Figure 3).
5. *Step 5:* Labeling of all PEs belonging to the zones P_1, P_2, P_3 and P_4 (Figure 8).
6. Step 6: Elimination of all PEs that do not belong to the zones P_1, P_2, P_3 and P_4. (Figure 8).
7. *Step 7:* Processing on the zones P1, P2, P3 and P4: this consists in determining PEs of the different quadrants NW, NE, SE and SW belonging to the convex envelope.

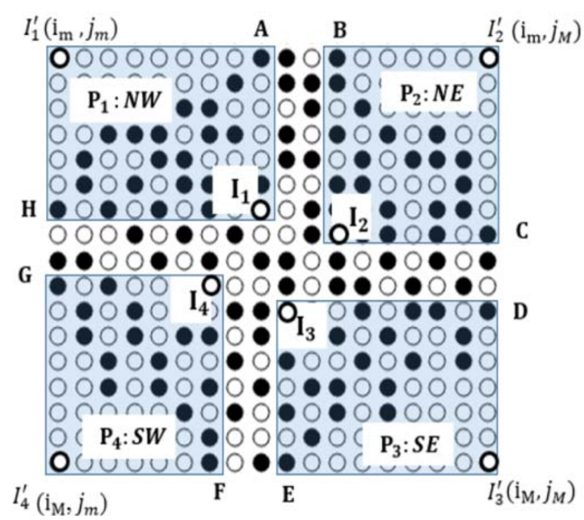


Figure 3: Step 1, Step 2 and Step 3.

Global Journal of Computer Science and Technology (F) Volume XXI Issue I Version I Year 2021

Step 1:

a) All the PEs of the image matrix get into Crosse Bridge (CB) (Figure 3) and perform in Ө (1) time [ELMES91]:

1. MIN operation for lines i. (i_m is the minimum of the lines).
2. MIN operation for columns j. ($j_m$being the minimum of the columns).
3. MAX operation for lines i. ($i_M$is the maximum of the lines).
4. MAX operation for columns j. (j_M is the minimum of the columns).

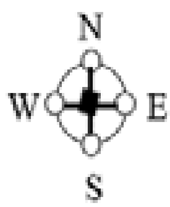


Figure 4 a: One PE in Crosse Bridge

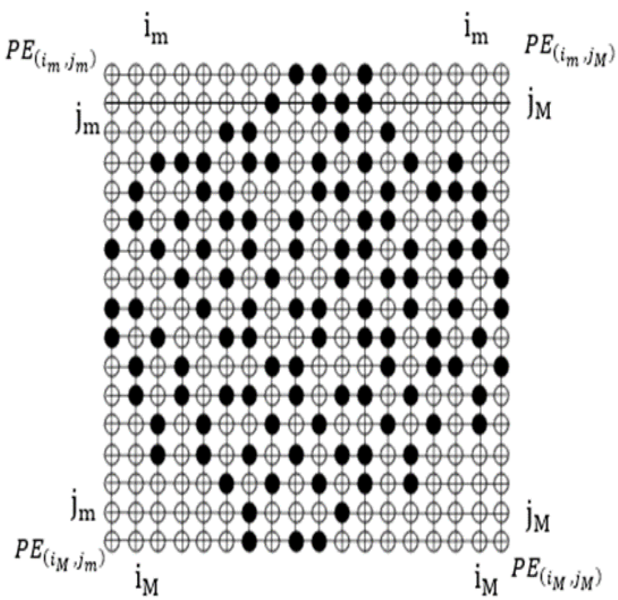


Figure 4 b: All the PEs of the image matrix in Crosse Bridge

Figure 4: Crosse Bridge

The (i_m, i_M, j_m, j_M) are all determined in Ө(1) time and the corresponding PEs are: $PE_{(i_m,j_m)}$, $PE_{(i_m,j_M)}$, $PE_{(i_M,j_m)}$, $PE_{(i_M,j_M)}$

b) All the PEs in each line go into Simple Bridge (SB) and make a MIN for the columns j then a Max for the columns j. We will have in each line at most two Marked PEs (PE_M). (Figure 5-a). For example, for the line i_m:

A = $j_{\min\_i_m}$ and $PE_A = PE_{(i_m,A)}$

B = $j_{\max\_i_m}$ and $PE_B = PE_{(i_m,B)}$

F = $j_{\min\_i_M}$ and $PE_F = PE_{(i_M,F)}$

E = $j_{\max\_i_M}$ and $PE_G = PE_{(i_M,E)}$

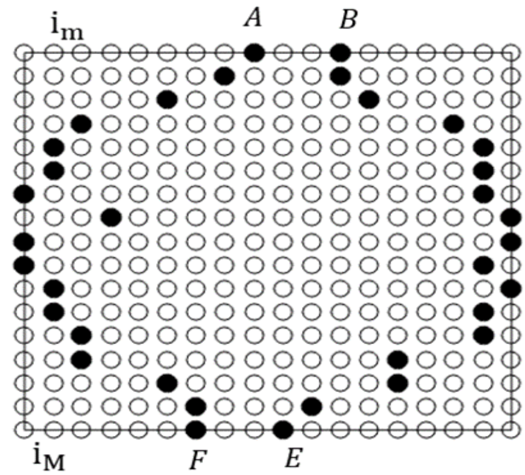


Figure 5 a: Min and Max lines

A = $j_{\min\_i_m}$,B = $j_{\max\_i_m}$

F = $j_{\min\_i_M}$,E = $j_{\max\_i_M}$

a) All the PEs of the image matrix get into Crosse Bridge (CB) (Figure 3) and perform in Ө (1) time [ELMES91]:

H = $i_{\min\_j_m}$ and $PE_H = PE_{(H,j_m)}$

G = $i_{\max\_j_m}$ and $PE_G = PE_{(G,j_m)}$

C = $i_{\min\_j_M}$ and $PE_C = PE_{(C,j_M)}$

D = $i_{\max\_j_M}$ and $PE_D = D_{(D,j_M)}$

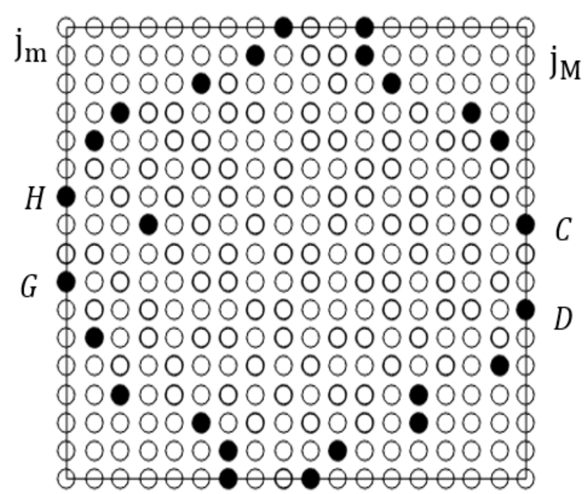


Figure 5 b: Min and Max columns

H = $i_{\min\_j_m}$, G = $i_{\max\_j_m}$

C = $i_{\min\_j_M}$, D = $i_{\max\_j_M}$

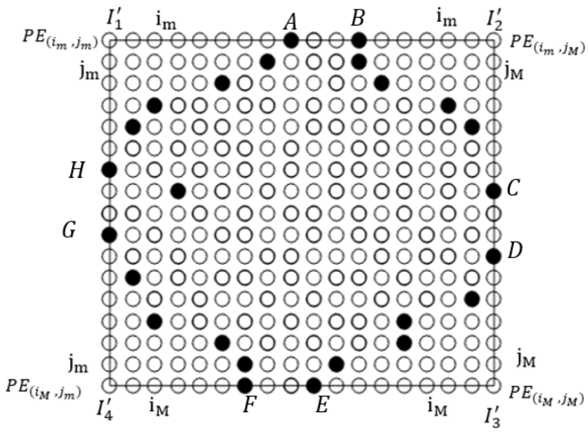


Figure 5: Summits: $PE_{(i_m,j_m)}$, $PE_{(i_m,j_M)}$, $PE_{(i_M,j_m)}$ $PE_{(i_M,j_M)}$et $PE_M : M \in \{A,B,C,D,E,F,G,H\}$. belonging to the sides of the quadrilateral encompassing the set of points of the image.

Step 2:

a) All PEs are configured in (SB) with respect to the line. (Figure 5).

b) The $PE_{(i_m,j_m)}$, $PE_{(i_M,j_m)}$, transmit on the bus on their right a code indicating their presence. All unmarked PEs will receive this code, those to the right of the two PEs having (i_m ,j_m) and (i_M , j_m).

c) The $PE_{(i_m,j_M)}$, $PE_{(i_M,j_M)}$ transmit on the bus to their left a code indicating their presence. All unmarked PEs will receive this code, those to the left of the two PEs having (i_m ,j_M) and (i_M , j_M) .

d) The operations indicated in (a), (b) and (c) will be done on the columns.

e) Result: 4 PEs will receive 2 codes one on their line and another one on the columns and will memorize them in I'_1, I'_2, I'_3 and I'_4. (Figure 5).

Step 3:

a) All the PEs of the image matrix put themselves in (SB) on their column (Figure 6.a).

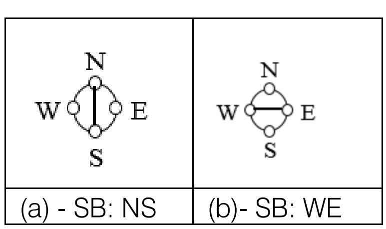


Figure 6: SB Rows and columns configuration (a) - SB: NS; (b)- SB: WE

b) The PE_A sends a code indicating the position of the column where it is located. So, all the PEs in this column will receive this code (Figure 7: quadrant P1).

c) All PEs in the image matrix are configured as (SB) on their line (Figure6-b).

The PE_H sends a code to all the PEs that are on his line. So, all PEs on this line will receive this code (Figure 7: quadrant P1). This operation runs for all others PEs:PE_B,PE_C,PE_D,PE_E,PE_F and PE_G.

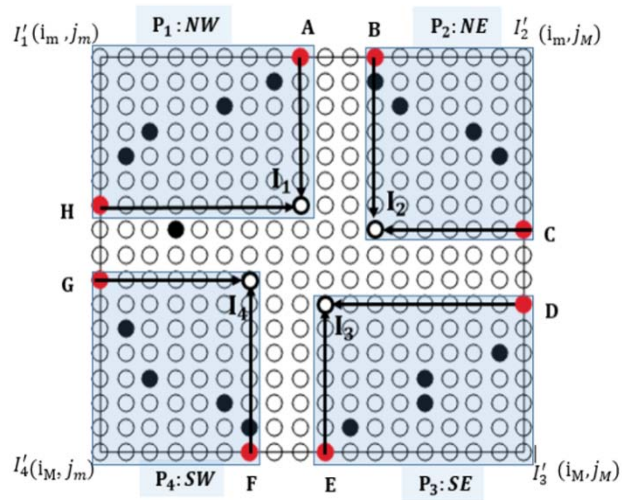


Figure 7: Determination of pointsI_1 , I_2 ,I_3 and I_4.

Step 4: Delimitation of the zones $P_1P_2P_3$and P_4: in the area P_1 (quadrant NW), all PEs that are between A and I_1 are in configuration (SB) on their column.

The PE_A gives them a code indicating that they have to block the bus that connects them with the PEs that are on their right (Figure 8.a).

In the same way PE_{I_1} communicates to the PEs that are in its line on the left, the code for disconnect PEs from the top line (Figure 8. a).

This operation will be executed in the zones P_2, P_3 and P_4.

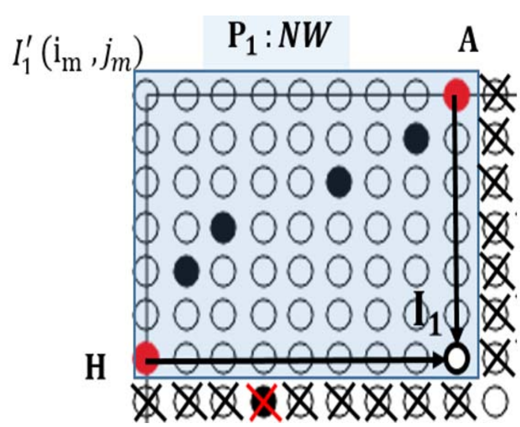


Figure 8 a: Delimitation of P_1 area.

Step 5: Labeling of the zones P_1, P_2, $P_3$and P_4: in zone P_1, since the PEs I'_1, A and I_1 are known, we proceed to the labeling all PEs belonging to P_1.

All the PEs in zone P1 are configured in Crosse Bridge (CB) state. The PE_{I_1} transmits an identifier code of the zone P1(Figure 8.b).

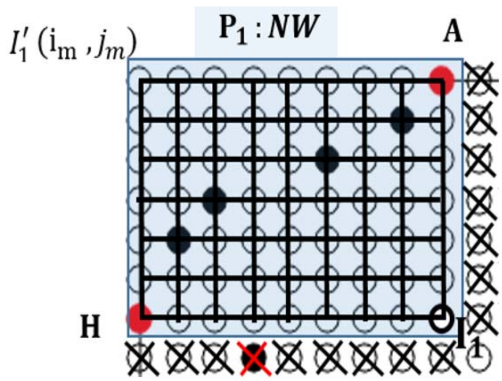


Figure 8b: Labeling of area P1

This operation will be executed in the zones P_2, P_3 and P_4 (Figure 8).

Consequence: All PEs in these areas are identified as part of the zones P_1, P_2, P_3 and P_4.

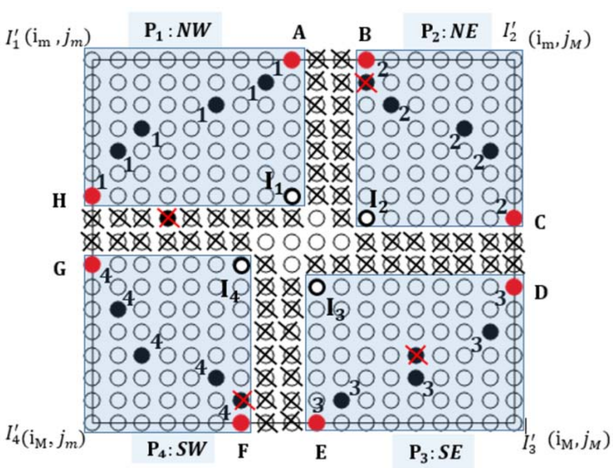


Figure 8: Delimitation and labeling of the zones P_1, P_2, P_3 and P_4.

Step 6: Elimination of all PEs that do not belong to the zones P_1, P_2, P_3 and P_4. (Figure 8).

a) All the PEs in the image matrix go into state (CB) except the PE I'_1.
b) The PEI_1' transmits to all PEs of the image matrix the cancellation order of marking if it is a marked PE. This implies that all PEs that do not belong to zones P_1, P_2, P_3 and P_4 will no longer be marked. Those who belong to these areas remain marked if they are.

Step 7: Treatment of labeled PEs belonging to zone P_i (Figure 9): Treatment of the area P_1 (Quadrant NW)

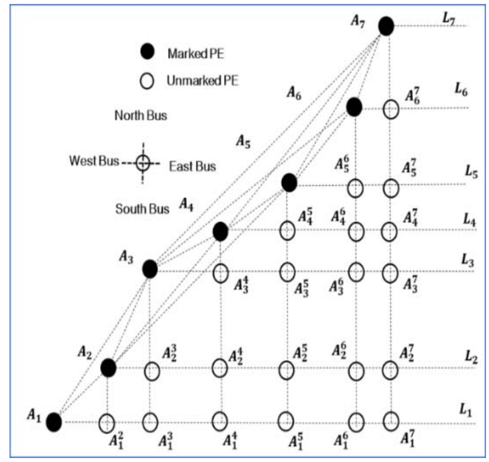


Figure 9: Treatment of the area P_1 (Quadrant NW)

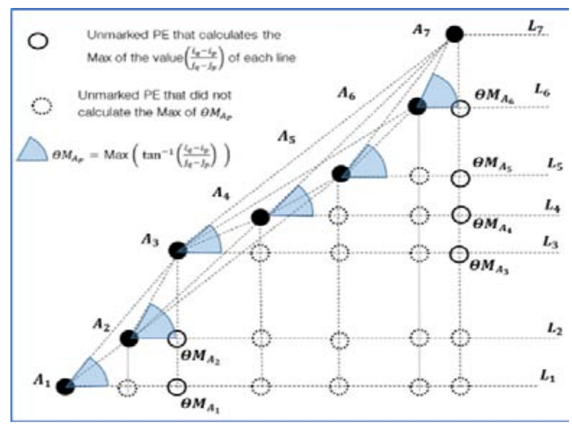


Figure 10: Determination of Max$\left(\frac{i_q-i_p}{j_q-j_p}\right)$

a) In each column only one PE marked, all others must be eliminated. Any Marked PE that has a labelled PE with an index lower than its index in the same column must be discarded. (Example, see Figure 8, zone P_3).
b) All unmarked PEs in the same column will execute the SB state except for only one remaining PE (A_7, A_6 , $A_5 A_4$, A_3 , A_2 and A_1).
c) Each marked PE transmits its position (i) on its south down column.
d) Each marked PE transmits on the East bus its position (j).
e) All the unmarked PEs A_p^q of row p and column q, which have received on the North and West buses the positions i_p and j_q, calculate the value $\left(\frac{i_q-i_p}{j_q-j_p}\right)$ in (A_1^2 ,.,A_1^7 , A_2^3 ,.,A_2^7 , A_3^4 ,., A_3^7 , A_4^5 ,..., A_4^7 , A_5^6 ,..., A_5^7 et A_6^7).

f) Each PE A_P^q calculate: $\left(\frac{i_q-i_p}{j_q-j_p}\right)$ or $\theta_P^q = \tan^{-1}\left(\frac{i_q-i_p}{j_q-j_p}\right)$, all the unmarked PEs of the line p execute the operation Max [ELMES91] in this line.

1. In the case of the line L_1 all PE_1(A_1^2 ,..,A_1^7) will run the Max's $\left(\frac{i_q-i_p}{j_q-j_p}\right)$ or θM_{A_P} = Max (θ_P^q) and we are going to have A_1^3 who will have the Max of $\left(\frac{i_q-i_p}{j_q-j_p}\right)$.
2. In the line L_2 the PE A_2^3 will have the Max.
3. In the line L_3 the PE A_3^7 will have the Max.
4. In the line L_4 the PE A_4^7 will have the Max.
5. In the line L_5 the PE A_5^7 will have the Max.
6. In the line L_6 the PE A_6^7 will have the Max.

g) All unmarked PEs, having already calculated θM_{A_P}, execute a Max [ELMES91] of the value $i = i_q - i_p$ on their column j_q.

Example:

1. In the line L_1, A_1^3 will be selected (it has the value $(i_3 - i_1)$ which is greater than $(i_3 - i_2)$)
2. In the line, A_3^7 will be selected (it has the value $(i_7 - i_3)$ which is greater than $(i_7 - i_4) > (i_7 - i_5) > (i_7 - i_6)$.
3. L_1, A_1^3 transmits through the open bus to the PE A_1^2 the order to eliminate the $PE_M(A_2)$ on its column.
4. L_3, A_3^7transmits through the open bus to all PEs $A_3^4A_3^5$ and $A_3^6$the order to eliminate all PE_M on their column. In the other lines, noPE_M will be selected.

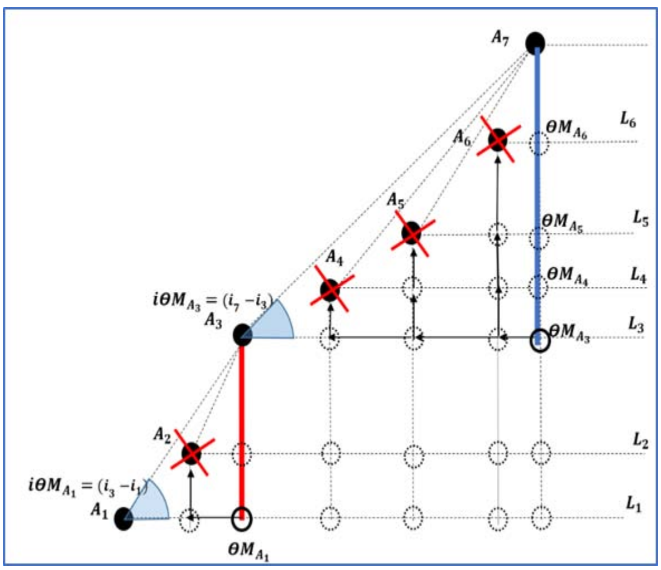


*Figure 11:*Max on the line index on the samecolumn

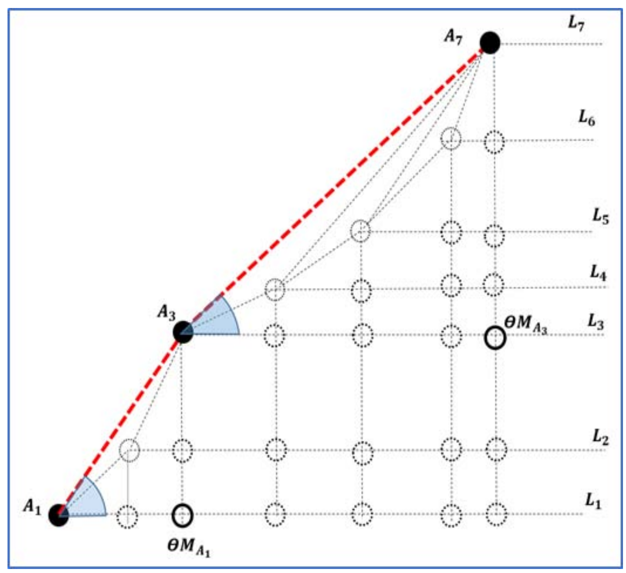


Figure 12: Result of the P1 zone The Marked PEs (PE_M)

Result: The Marked PEs ([PE]_M) : A_2 , A_4 , A_(5) and A_(6) are eliminated, the [PE]_M A_1 , A_3 and A_(7)belong to the convex hull of the area

IV. Conclusion

This paper has dealt with a parallel algorithm of determining the convex hull of a two-level 2D image with a complexity ϴ (1) time. This is executed on a parallel machine (RMC), of size n x n Elementary Processors. The algorithm is essentially based on projections and the calculation of Min / Max in ϴ (1) time. In a future work, we will extend this algorithm to 3D space and always in ϴ (1) time.

References Références Referencias

1. [FU97]: FU A. M. N., and YAN H. *Effective classification of planar shapes based on curve segment properties*, Pattern Recognition Lett. 18 (1997), 55-61.
2. [CHO81]: CHOW P., *A parallel algorithm for determining convex hulls of sets of points in two dimensions*, in proc.19th Allerton Conf. Common., Contr., Comput., 1981, pp. 214-233.
3. [NAT81]: NATH S. N., MAHESHWARI N., and BHATT P. C. P., Parallel algorithms for the convex hull in two dimensions, in Proc. Conf. Anal. Problem Classes Programming Parallel Comput., 1981, pp. 358 -372.
4. [ALD83]: ALD M., *Parallel algorithms for convex hulls*, Dep. Comput. Sci., Queens Univ., Kingston, Ont., Canada 1983.
5. [KIM87]: KIM C. E., and STOJMENOVIC I. Parallel algorithms for digital geometry, CS-87-179, Washington State University, Pullman, December 1987.

6. [LIN93]: LING T., et al., Efficient Parallel processing of image contours, IEEE Trans. Pattern Anal. Mach. Intelligence 15(1) 1993, 69-81.
7. [PRA89]: PRASANNA V.K., and REISIS D.I. : Image computation on meshes with multiple broadcast, IEEE Trans. Pattern Anal. Mach. Intelligence 11(11) (1989),1194-1201.
8. [MIL85]: MILLER R., and al.: *Geometric algorithms for digitized pictures on a mesh connected computer*, IEEE Trans. Pattern Anal. Mach. Intelligence 7(2) (1985).
9. [LI89]: LI H., MARESCA M., *Polymorphic Torus Architecture for Computer Vision*, IEEE Trans. on PAMI. Vol.11, n°3, pp. 233-242, March89.
10. [HAY98]: HAYACHI T. et al.: An (O (log log n)^2) time algorithm to compute the convex hull of sorted point on a reconfigurable meshes, IEEE trans. Parallel Distrib. Sys. 9(12) 1998, 1167-1179.
11. [ERR05]: ERRAMI A., KHALDOUN M., ELMESBAHI J., and BOUATTANE M., Θ (1) time algorithm for structural characterization of multi-level images and its applications on a reconfigurable mesh computer, Journal of Intelligent and Robotics Systems, 2005.
12. [BOU02]: BOUATTANE O., ELMESBAHI J., ERRAMI A., *A fast parallel algorithm for convex hull problem of multi-leveled images*, Journal of Intelligent and Robotic Systems 33: 285-299, 2002.
13. [ELMES86]: J. Elmesbahi and J. Cherkaoui, Structure Analysis for Gray level Pictures on a mesh connected computer, Proc. of IEEE Internat. Conf. on SMC, 10 (1986), 1415-1419.
14. [ELMES91]: J. El Mesbahi, Θ (1) algorithm for image component labeling in a mesh connected computer, IEEE Trans. Syst. Man Cybern., 21 (1991), 427-433.
15. https://doi.org/10.1109/21.87089.