International Journal Of Advance Research In Science And Engineering http://www.ijarse.com
IJARSE, Vol. No.2, Issue No.4, April, 2013 ISSN-2319-8354(E)

ENHANCED WAY FOR COMPUTING SMALLEST CONVEX POLYGON AND EULER NUMBER

Rishika Agarwal[1], Sonal Agarwal[2], Neelesh Mishra[3]

UG, [1,2]Department of Computer Science Engineering,
[3]Assistant Professor Department of Computer Science Engineering,
Raj Kumar Goel institute of technology for Women, UP (India)
Gautam Buddh Technical University, Lucknow, (India)

ABSTRACT

A convex hull is a polygon which encloses all given set of points.Euler number or Euler characteristic of an image has proven to be an important feature in many image analyses and visual inspection applications. This paper presents an algorithm for fast computing the convex hull of a planar scattered point set, which pre-strike an initial convex hull boundary, remove internal points in the boundary which cannot be the minimum convex hull vertex points, improve the efficiency of computing the convex hull. Experimental analysis and comparison show that the algorithm has better time complexity and efficiency of obtaining the minimum convex hull. We present an algorithm for finding the convex hull of a sorted planar point set. This algorithm uses graham-scan for recognizing those points computing the smallest convex polygon and bresenham method to represent the computed smallest convex polygon.

In this paper, we have also calculated the Euler number by using the concept of convexity and concavity. Convexity (or the convex number) $N_2^+(X, \alpha)$ is defined as the number of first entries in a given direction α.Concavity (or the concave number) $N_2^-(X, \alpha)$ is defined as the number of exits in a given direction α.

Keywords: Bresenham Method, Convex Polygon, Euler Number, Convex Hull (CH).

I INTRODUCTION

Euler Number is a topological parameter, which describes the structure of an object, regardless of its specific geometric shape. Various formulas have been developed for calculating it. Here the Euler Number is calculated with the help of convexity, concavity and diagonals. Euler characteristic of an image has proven to be an important feature in many image analyses and visual inspection applications. EulerNumberfor any graph can also be calculated by the following formula:

$$|\mathbf{V}| + |\mathbf{F}| = |\mathbf{E}| + R$$

where**V** are thevertices, **E** are the edges and **F** are the faces. Furthermore Euler Number of a binary image can be computed in terms of the number of terminal points (points with just one neighbor) and the number of three-edge points (points with only three neighbors) of the skeletons of the regions inside the image.[6]

More in detail, if *Tps*is the number of terminal points of the *n* skeletons from *n* objects in the image, and *TEps*is the number of three-edge points of these *n* skeletons, then

E= (*Tps*- *TEps*)/2

Euler Number in figure given below is calculated by the difference between the connected components and holes, where white color represents the holes and black represents the number of connected components.[5]

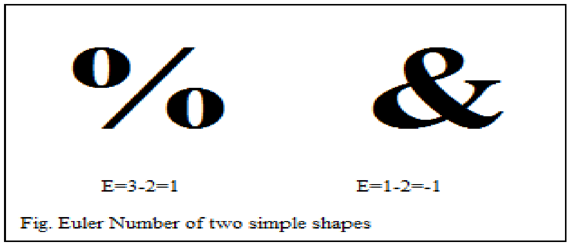

Fig1.1 Euler Number of Two Simple Shapes

Finding the convex hull (CH) of point sets is a fundamental issue in computational geometry, computer graphics, robotics etc. Some of the most popular algorithms of building CHs include Graham scan [1], Jarvis march [2], Monotone chain[3], Quickhull [4], Divide–and–Conquer [5] and Incremental [6].A convex hull is a polygon that encloses a given set of points with minimum perimeter nodes. The convex hull of a set of points *S* in *n* dimensions is the intersection of all convex sets containing *S*. For *N* points *x*1, ,*xN* the convex hull *C* is then given by the expression :

$$C=\sum_{j-1}^{N} \lambda j\, Xj : \lambda\text{j}>=0 \text{ for all j}$$

The intuition is fairly straightforward. For two explanatory variables, the convex hull is given by a polygon with extreme data points as vertices. There are two main properties of convex hulls that should be explored before introducing Graham's Scan or Jarvis' March. [2].

II RELATED WORK

There are lot of papers and algorithms that have been presented for computing Euler number and convex hull.

Some of the popular algorithms for Euler number have presented by William K.Pratt.[1], Tom Davis[6] and Hexar [7]. Some of the most popular algorithms of building Convex Hulls include Graham scan [1], Jarvis march [2], Monotone chain[3], Quick hull [4], Divide–and–Conquer [5] and Incremental [6].

2.1 Euler Number

The Euler number is computed by calculating the difference in convexity and the concavity of the image [1]. A region **R** is said to be convex if, for every x_1, x_2 in **R,** [x_1, x_2] belongs to **R.**A convex hull for a region **R** is defined as the convex closure of region **R.**In this we have considered the 4-connectivity as well as 8-connectivity. Convexity, concavity and diagonals for an image can be calculated by a 2*2 mask for image with 4-connectivity [5].

Concavity is calculated through mask with three 1's and single 0. Convexity is calculated through mask with three 0's and single 1 [5].

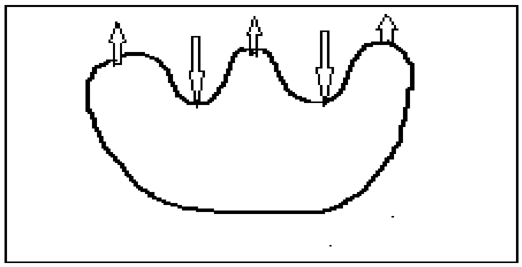

Fig.2.1 Calculating Convexity And Concavitity

Convexity is also defined as the number of enteries and concavity as the number of exits. Arrows in the upward direction are the exits and the arrows in the downward diraction are the enteries.

Upstream convexity:

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

UpstreamConcavity:

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

To find the Euler number from matrix refer for reference number of [1], [2], [3], [5], [6], [10] and [11].

2.2 Convex Hull

Convex hull was proposed to be implemented by Graham Scan or by Jarvis' March properties.

One of these properties is that all of the points in the final polygon must be indented outwards, or more formally, convex. If any points are concave, there exists a better path by excluding that point, and drawing a direct line between its two neighbor points, which reduces the overall perimeter point count by one, as can be seen in the figures below. This property is exploited by Graham's Scan. [4]

Another important property is that the most extreme point on any axis is part of the convex hull. This fact is apparent if you consider an example in which an extreme point is not in the convex hull. This would mean the perimeter of the convex hull passes through a point less extreme than that point. However, it is then obvious that the extreme point would not be included in the enclosed set, thus avoiding a fundamental characteristic of convex hulls. This property is used by a number of algorithms because they rely on information computed from an initial point in the convex hull. [8]. In this paper we are showing an enhanced method that uses bresenhalms' algorithm [7].

The Bresenham / Midpoint Algorithm is a solution that not only correctly picks the points, it does it with integer math only, which results in a big speedimprovement. This algorithm, of course, relies on some assumptions such as:

· Non-infinite slope

· $x1< x2$

· $y1 < y2$

For examples references are in [7], [9] and [11].

Recently, several novel algorithms are developed to obtain CH for point set: Frank and Matouk [12] present a polynomial-time algorithm for the D-convex hull of a finite point set in the plane. In [10], new properties of CH are derived and then used to eliminate concave points to reduce the computational cost.

III PROPOSED WORK

A new algorithm is proposed so as to reduce the computation time and complexity as compared to previous algorithms. In this paper, two algorithms have been proposed. One is for Euler number and the other is for computing convex polygon.

3.1 Euler Number

The algorithm starts with taking image as input in raw form. Image can be inputted in two modes- read and write. In read mode the image could only be seen but in write mode we can perform various operations also. We are taking image in the form of a raw image and the memory is also allocated for it. An image is composed of all the flat connected regionsrepresenting the projections of the objects onto thediscrete plane. Every pixel of image has some or the other value. So the convexity, concavity and diagonals are calculated at every pixel of the image.Diagonals are calculated through mask with alternative 1's and 0's.Without loss of generality, only the case for squared pixels is presented**.**The first pixel with value '1' in the image should be upstream convexity. Hence this convexity refers to the connected component. Now any pixel with the mask as upstream concavity canbe a Hole or Background.

The Euler Number is then calculated by the help of the formula specified in the proposed algorithm.

Proposed Algorithm:

Input: Image in raw form

Output: Euler Number of the image

1. Read the image given.
2. Find theconcavity, convexity and diagonals values from the image
3. Calculate concavity, convexity and diagonals in the image.
4. Calculate Euler number as:
   E= ((convexity-concavity) +(2*diagonals))/4.

3.2 Convex Hull

The first phase of the algorithm is to find the leftmost point which has smallest x and y coordinates. Once the point has been located, the remaining points are sorted in increasing angular value. After this, the main work of graham scan starts. Graham Scan computes convexity of the point to check whether it is a part of convex hull or not. If the point is convex, the algorithm adds it up to the array and proceeds to the next. It the point is not convex, it removes the current point or the topmost point from the stack. The algorithm needs to backtrack to the previous points to see if the changes caused affect other nodes to become convex or not. The process continues until all the nodes are traversed. The functions used to compute convex hull are leftmost(), sort(), make_hull, swap(), nonleftturn().

The leftmost() function is used to calculate the leftmost point among the set of input coordinates. The sort() function arranges the coordinates according to their increasing angular value. Swap() is used for swapping of values and positions. Nonleftturn() returns a Boolean value to determine whether the three selected coordinates make a non-left turn or not. Make_hull() finally computes the points of the convex hull and stores it in a stack.

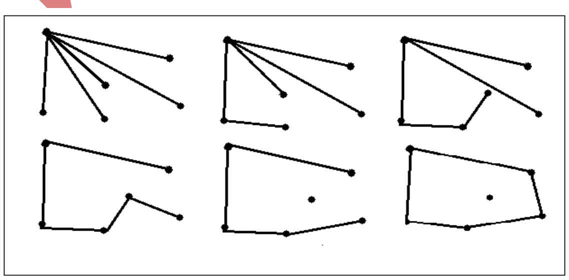

Fig. 3.1 Computing Convex Polygon

The points are stored in array after been read from the text file. These points are sorted in increasing angular value except the leftmost point.Graham's Scan continuously removes and adds points to and from the stack. Hence, due to the variable number of points, double link list is used which is also circular. This allows us to use the concept of dynamic memory and also facilitates us in a way that when a point is found to be convex, the node can simply be deleted and previous and next nodes can be adjusted accordingly to point each other according to the changes. Now, after all the points in the convex hul are identified, we have used bresenhalm algorithm to join these points.

Finally the image is written as a raw file.

Proposed Algorithm:

Input: A set of points in a text file

Output: Smallest convex polygon

1. Read the file to get the set of points and store it in an array.
2. Initialize a pointer image that will give you the resultant convex hull output.
3. Create a stack.
4. Find the leftmost point in the array.
5. Swap the leftmost points with the a[0] points in array.
6. Sort the array but leaving a[0] elements unchanged.
7. Push the top 3 elements of the array in the stack.
8. for(I =3; I <NoOfPoints_array; I ++)
9. {
10. Let s, t be the top two points in the stack.
11. IF a[i] makes a non left turn with respect to s, t then pop the top element.
12. ELSE push a[i] into the stack.
13. }
14. //points in the stack are the convex hull points.
15. for(I =0; I <NoOfPoints_hull; I ++)
16. {
17. Let s, t be the top two points in the stack.
18. Make_Hull (&s, &l , image).
19. }
20. Display output as an image raw file.

IV IMPLEMENTATION

The implementation graph of our algorithm shows us that it is better and faster than other implemented algorithms. The running time of our proposed algorithm is much lesser than others.

The original data points are created randomly. For each size of points set, the mentioned three algorithms are tested. The running time of finding the convex polygon for different size of points set uniformly is compared separately. We have implemented and compared our algorithm with the two popular algorithms Monotone chain and Jarvis March.

The running time of different algorithm has been compared with proposed algorithm. Now the comparison time is below:

Size of points	Running time of CH algorithms(/s)		
	Monotone chain	Jarvis March	Proposed algorithm
10	0.256	1.33	0.105
20	0.558	3.380	0.220
30	1.079	4.46	0.458
40	1.720	4.826	0.89

Table 4.1: Running Time Comparison of CH Algorithms

The comparison graph shows that proposed algorithm takes minimum time to execute that helps of finding better result in image analysis.

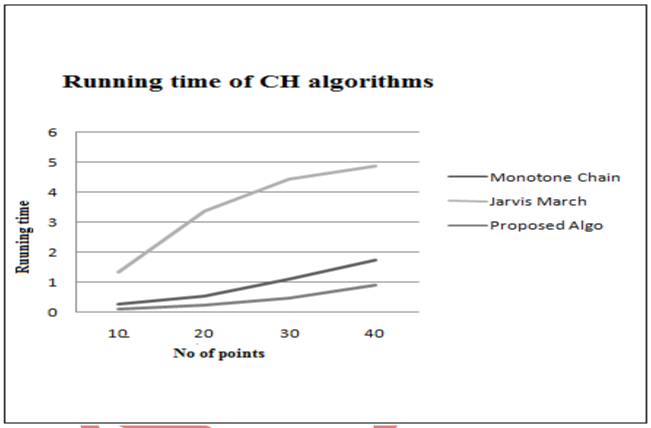

Fig 4.1 Comparison of Different CH Algorithm

V CONCLUSION

This paper presents an enhanced way of computing smallest convex polygon and Euler number of a given shape.

Computation of Euler number is based on information directly obtained from each pixel of all shapes in the image. Due to the fact that each pixel can be processed independently, the proposal can be parallelizable. The algorithms related with 8-connected regions will be the subject of future research. For objects with no holes, we can also calculate the number of connected components.The computation of convex polygon requires computing of internal points. Compared with the popular CH algorithms, the algorithm is faster than Monotone chain and Jarvis march. The shortcoming of this algorithm is that the space cost is expensive due to that several intermediate variables need to be allocated for sorting and storing the sorted sequence.

REFRENCES

[1] Digital image processing, Convex Hull, William K.Pratt.

[2] Digital Image processing, Convex Hull, Rafael C. Gonzalez.

[3] Digital image processing, Euler Number, Anil K.Jain.

[4] Introduction to algorithms, Graham Scan, by Thomas H.Cormen.

[5] Alternative Way to Compute the Euler Number of a Binary Image by J. H. Sossa-Azuela, E. B. Cuevas-Jiménez , D. Zaldivar-Navarro

[6] Euler's Theorem ,Tom Davis

[7] Graphics Tutorial, Bresenhalm Algorithm, Hexar.

[8] An investigation of Graham Scan, Chris Harrison

[9] Convex Hulls: Complexity and Applications (A Survey), SuneetaRamaswami (University of Pennsylvania)

[10] S. B. Gray, Local properties of binary images in two dimensions, IEEE Trans. Computer 20(5), 551-561 (May 1971).

[11] Ch. R. Dyer, Computing the Euler number of an image from its quad tree, Computer Vision, Graphics Image Process: 13, 270-276, (1980).

[12] V. Frank and J. Matouk, "Computing D-convex hulls in the plane", *Computational Geometry*, vol. 42, pp. 81-89, 2009.