International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Minimum Bounding Circle of 2D Convex Hull

Asia Mahdi Naser Alzubaidi

Computer Science Department, College of Science, Karbala University, Karbala, Iraq

Abstract: ***In mathematical, minimum enclosing circle problem is a Geometrical issue of calculating the smallest circle that contains all of a finit set of points in the Euclidean plane. The problem of finding the minimum circuler container can ubiquitous in diverse set of applications such as collision avoidance, hidden object detection and in planning the location of placing ashared facility like a hospital, gas station, or sensor devices etc. moreover, The usefulness of minimum containers occurs in a variety of industrial applications like packing and optimum layout design. The algorithm can be applied to many other fields, ranging from a straightforward consideration of whether an object will fit into a predetermined of circuler container, or whether it can be made from standard sized stock. In this paper, we describes a method for determining the minimum bounding ball of a set of 2D convex polygon based on Chan's algorithm which consist of graham scan with Jarvis march methods. The suggested method involves of three steps. Firstly, generates any number of 2D points set using randomly function. Secondly, we compute the convex hull for points set depend on Chan's algorithm. the third step of proposed scheme includes finding the minimum enclosing circle of convex polygon. the experimental analysis and results of presented algorithm find that the computational time is significant for any number of vertices of 2D points set.***

Keywords: 2D points; minimum bounding ball; convex hull; graham scan method; Jarvis march algorithm

1. Introduction

This paper describes routine for calculating and updating the smallest enclosing circle of a finite point set. mathematically, the minimum enclosing circle is the outline of the closed disk of smallest area covering the point set and we usually recognize the disk with its bounding circle[1].Actually, the problem of determining the minimum bounding ball for set of objects in m-dimension has been developed and adopted for use in many areas such as in computational geometry for example in collision detection of objects or in collision avoidance, hidden object detection, in finding area of object, and in an approximation object in spatial indexes like the R-Tree and its alternates. Also, it can be used in industrial applications likes packing and optimum layout design and it founds in 2D applications of mathematics likes calculate the placement of an emergency facility for a set of customer modeled as demand points[2]. In this paper we give an insight for computing the minimum bounding ball of 2D convex polygon. So, the proposed algorithm began by generate the 2D points set and then construct the convex hull polygon using Chan's approach to determine the boundary of finite point set .The final step of proposed scheme includes finding the minimum bounding ball of convex polygon. This introduction gave a brief outline over the bounding circle algorithm and the rest of this paper is arranged as following: After a discussion of related backgrounds and provides some information about the algorithm's history in Section 2, section 3 discuss the Chan's convex hull algorithm. The actual bounding ball algorithm will be presented in section 4. Section 5 gives the experimental results of suggested algorithms with test examples and section 6 provides the paper's conclusions and future works.

2. Literature Survey

The minimum bounding circle problem was study in widely manner for instance Asano et.al present an algorithm for computing a shortest path between to vertices in a square grid graph with edge weights that uses memory less than linear in the number of given points can be tailored to work in a read-only environment with O(log n) extra space[3].In 1012 some researchers propose a general prune-and-search technique in read-only environment and others to improve the time and space complexity of solving the minimum bounding circle problem in linear time with O(1) extra space[4]. While paper[5] describes the minimum bounding circle problem and made comparison between Naive Approach and Welzl's Algorithm corresponding to space and time complexity they found that naive algorithm solves the problem in time O(n4).while, Emo Welzl in 1991Solves the smallest enclosing disc problem in expected O(n) time.

3. Convex Hull via Chan's Algorithm

The convex hull of any 2D points set or polygon in the plane can be defined as the smallest subset of most extreme points on X-axis or Y-axis which is not included of any interior set points. Mathematically, a polygon is convex if any line segment joining two points on the boundary stays within the polygon [6]. As an example in figure (1)

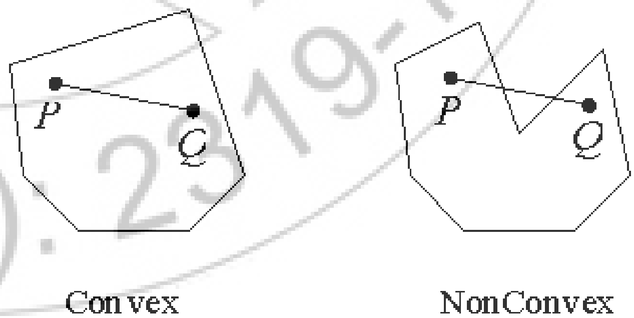

Figure 1: convex and non- convex polygon

Chan's algorithm is one of an optimal output sensitive algorithms that used to construct the convex hull of a set P of n points in 2D or 3D dimensional space [7]. In the planar case, an algorithm known as Graham's scan achieves in O(nlogn) running time. There is another algorithm known as a Jarvis March or gift wrapping algorithm, It has a complexity of O(h n) where h is the number of points in the resulting hull[8]. Chan's algorithm involves a very clever combination of the previous two algorithms, that are, Graham's scan and Jarvis's March, to form an algorithm that is faster than either one with O(n log h) time complexity. The main idea is smart and elegant based on Divide and conquer concept [9]. Divide stage begins by splitting the input points n in to r = n/m groups, each of size m range from 4,16,32,256,512,…, then for each subset we compute the sub hull p(m) using Graham's scan algorithm. While, in conquer or merge Stage we follow the general outline of Jarvis's march, starts from finding the smallest point of p0, then successively find the convex hull vertex that follow p0 in increasing order using binary search until we return Back to the smallest point again [10]. As shown in figure(2).

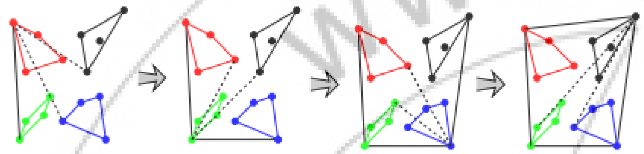

Figure 2: execution steps of Chan's algorithm

Chan's Algorithm (p)
1. For t = 1; 2; …
2. Let m = min(2^(2^t), n)
3. Let r = ceil(n/m)
4. Divide stage
5. For k = 1 to r do
6. For i = 1 to m do
7. Compute sub hull P(m)) using Graham's scan and store the vertices in an ordered array S in CCW oriented.
8. End
9. Merge stage
10. Find leftmost vertex S0 from all sub hulls that resulted from Graham scan algorithm
11. Compute final convex hull ch(h) using Jarvis March algorithm

3.1 Convex Hull by Graham's Scan Algorithm

The idea of algorithm is to find an extreme point p0 which is guaranteed to be in the resulting convex hull, then sort all other points as viewed from that vertex according to the angles in increasing order. Construct the convex hull by checking the direction(CCW) of point and adding the vertices when it make a left turn direction, and back tracking when making a right turn[10].

Graham Scan Algorithm (p)

1. Compute polar angles as follow, then sort the points according to their angle in CCW oriented.
- Find angle between the current line(x1,y1,x2,y2) with X-Axis.
- Let DX = (x2-x1) and DY= (y2-y1)
- If (DX = 0) and (DY = 0) return 0.
- Else compute angle by using angle =ArcTan2(DY,DX)*57.30.
- If (angle < 0) return angle +360.

2. Let p0 be the point in the p with minimum.
3. Let p1,p2,..,pm be the remaining points in p, sorted by polar around p0.
4. Push(S,p0) , n=1
5. for i =2 to m do
6. find direction (CCW) of three points (p0,p1,p2) as follow:
- dx1 = p1.x - p0.x
- dy1 =p1.y - p0.y
- dx2 = p2.x - p0.x
- dy2 = p2.y - p0.y
- if (dx1*dy2 > dy1*dx2) turn left
- else if (dx1*dy2 < dy1*dx2) turn right

7. while (CCW (p [n-1],p [n], p [i]) > 0)do
- if (n > 1)then n = n- 1
- else if (i = m) break
- Else
- go to 13

8. End while
9. n = n+1
10. Swap(p [n] , p [i])
11. Push(S, p)
12. End for
13. return S

3.2 Convex Hull by Jarvis March Algorithm

This algorithm manipulates the points on the convex hull in the order in which they appear. it is start with an extreme point p of a point set S which represent the leftmost point. At each step, we test each of the points, and find the one which makes the smallest turn right l and has to be the next one on the hull. Then we update p to l and repeat the process until we end up with the leftmost point [11]. As indicated in figure (3).

Jarvis March (S)

1. Let S0 the point whose x-coordinate is the smallest, to be the first point in the output convex hull ch
2. Let L be the index of point S0 in S
3. Repeat
4. h= 0
5. ch[h] = S0
6. h=h+1
7. Compute farthest point q =(p+1)% n
8. For i = 0 to n - 1
9. if CCW (S[S0],S[i], P[q])= turn right
10. q = i
11. S0 = q
12. until S0 =L
13. Return ch

4. Minimum Bounding Ball

Actually, in 2D space a minimum bounding ball of a finite linear set of points resembles to the algorithm of computing the simplex shape of linear programming. The idea behind the algorithm can be imagined as a balloon strictly containing all the generated randomly points and then deflate it until it can't shrink anymore without losing of any point[12]. The bounding ball B of a linear geometric objects set S and specified by a center point C and a radius R also called the "minimal spanning sphere" of the objects can be computed efficiently with running time O(n).figure(4) show some steps of minimum bounding ball algorithm.

Minimum Bounding Ball Algorithm
Input: linear geometric objects set S.
Output: B represents the Minimum Bounding Ball of S.
Begin
1. a good initialization for a bounding ball B is made by calculation two points of S that are remote from each other by selecting ones on opposite extremes of the bounding box for S, and using the line between them as an initial value of diameter, then, the center of diameter is the initial ball center C, and half the length of the diameter is the initial ball radius R.
2. each point P of S is tested for presence in the current ball this is done by simply checking its distance from the center C is less than or equal to the radius R. If the next point P_{k+1} is in the current B_k, then $B_{k+1}=B_K$ and one just proceeds to the next point. while if P_{k+1} is outside B_k, then B_k is expanded just enough to include both itself as well as the point P_{k+1}. This is done by drawing a line from P_{k+1} to the current center C_k of B_k and extending it further to intersect the far side of Bk. This line is then used as the new diameter for an expanded ball B_{k+1}. As shown in figure(6), it clearly contains the prior ball B_k and all points of S already considered, and no additional recursion is needed[13][14].

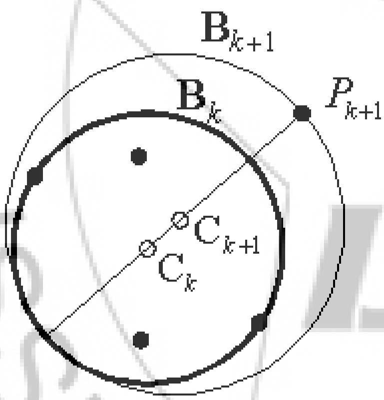

Figure 4: Minimum Bounding Ball

3. After we having R and C from previous step using any algorithm to draw circle shape, in this paper we use polar method as following:
Polar Algorithm
Input: C center point, R radius.
Output: draw circle shape.
Begin
1. Two_pi=2*pi.
2. Theta=0.
3. Dtheta=1.0/R.
4. While Theta<= Two_pi
- $X= C_X + R* \cos (Theta)$.
- $Y=C_Y + R * \cos (Theta)$.
- Plot (round(X), round(y)).
- Theta= Theta + Dtheta
5. End while.
Finish

5. Experimental Analysis and Results

In this section, more details can be found of system software implementation. We explain the results of software operation with an example. The first step of system begins by generating randomly points set on 2D space, this set elected to construct the convex hull polygon. Figure (5) represents the GUI of points generation with 100 number distributed randomly.

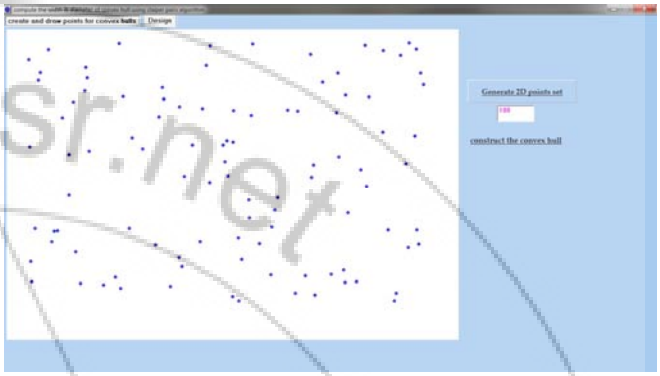

Figure 5: GUI of randomly points

After having the 2D points we sort these points relatively to x-coordinate value and then, applying Chan's algorithm to construct the convex hull. This algorithm hybrid between graham scan and Jarvis march methods. Figure(6) shows the GUI after applying graham algorithm and as shown the number of points became 45.

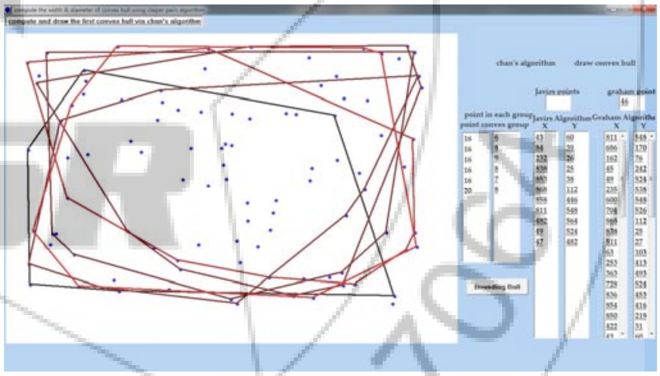

Figure 6: GUI of graham algorithm

Building of Chan’s convex hull using graham scan to construct n sub groups then applying Jarvis march algorithms to obtain one group represent the final convex hull as depicted in the GUI Chan’s algorithm in figure(7) with their minimum enclosing circle in red color. Basically, from our implementation of system we could see that the Chan's algorithm could deal with any number of polygon points in the same effectiveness since it depend on divide and conquer algorithm to construct the convex hull in O(n log n) running time. From our example the input vertices was 100 points while the output data after applying Chan's algorithm equal to 11 points

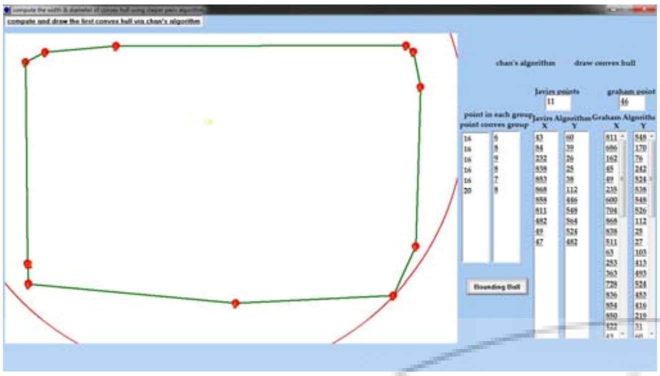

Figure 7: GUI of Chan's Algorithm

Finally, Figure (8) shows the GUI of minimum bounding circle of example with 5 points set.

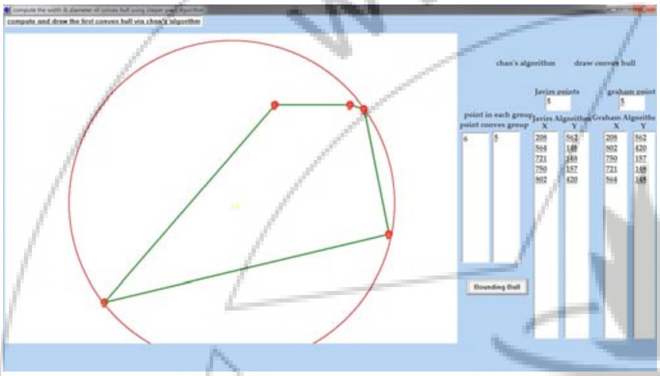

Figure 8: minimum bounding circle

6. Conclusion

This paper describes an alternative method for determining the minimum bounding ball of finite 2D points set. The algorithm works in an incremental manner and implemented as a semi-dynamic data structure, thus allowing to insert points while maintaining the diameter among the two farthest points. Therefore, the suggested above algorithm can be used for calculating the minimum bounding containers of any 2D points in fast, robust, and high efficiently manner.

7. Future Works

Proposed system provides good results.Therefore, the working and developing on it is very visible. We can suggest some future works:

- We could developed the proposed algorithm by using sets of circles instead of points in 2D dimension plane and then determine the minimum bounding circles of them.
- One of interesting problem is to design a sub linear-space routine for computing the shortest bounding sphere on the Delaunay triangulation of a point set in the plane.

Acknowledgment

We would like to thank anonymous referees for their constructive comments.

References

[1] M.Planck Institute for Informatics "CGAL Basic Library Reference Manual/ available at http://graphics.stanford.edu/courses/cs368-00-spring/TA/manuals/CGAL/ref-manual2/Optimisation/Min_circle_2.html,Jan-2000.

[2] L.B.roth ,"optimal containment ",doktor dissertation, technische universit¨at m¨unchen zen rum mathematic, des-2009.

[3] T. Asano, B. Doerr,"Memory-constrained algorithms for shortest path problem", In CCCG, 2011.

[4] M.De, S.C.Nandy , S. Roy " Minimum Enclosing Circle with Few Extra Variables", Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), pp. 510–521, 2012.

[5] J.Adarsh, S.Sahai," The Smallest Enclosing Circle Problem", Indian Institute of Technology, Kanpur, November-2013.

[6] D.Sunday,"The convex hull of a planer point set", available at http://geomalgorithms.com/a10-_hull-1.html ,2012.

[7] Wikipedia,M.Timothy,'' Chan's algorithm'', available at http://en.wikipedia.org/wiki/Chan's_algorithm ,2013.

[8] M.Hoffmann,"Chan's Algorithm", available at http://www.ti.inf.ethz.ch/ew/courses/CG09/materials/v2.pdf ,2009.

[9] T.Switzer," 2D Convex Hulls: Chan's Algorithm", online available at http://tomswitzer.net/2010/12/2d-convex-hulls-chans-algorithm/ ,2010.

[10] Z.Bo, H.Xisheng, J.Mingxun,"Convex Hull in 2D ",available at http://www.tcs.fudan.edu.cn/rudolf/Courses/Algorithms/Alg_cs_07w/Webprojects/Zhaobo_hull/ ,2007.

[11] M.D.Mount,"CMSC754ComputationalGeometry", Department of Computer Science University of Maryland,2007.

[12] K.Fischer,"Smallest enclosing balls of balls",Swiss Federal Institute of Technology, Doctor of Technical Sciences dissertation, ,2005.

[13] J.Ritter, "An Efficient Bounding Sphere" In Andrew Glassner ,editor, Graphics Gems,chapter 5,p301-p303,AP Professional,Bosten,1990.

[14] B.Gartner, "Approximate Smallest Enclosing Balls",Available at http://www.ti.inf.ethz.ch/ew/lehre/ApproxGeom08/notes/05-Smallest_Balls.pdf , Lecture Notes for Approximate Methods in Geometry ;esp Chapter 5,2008.

Author Profile

Asia Mahdi, Awarded her B.Sc, and M.Sc, at University of Babylon, College of Science, Department of Computer Science in 1997 and 2002 respectively. She is an lecturer at Karbala University, Collage of Science, Computer Department. Here research interests include: Computational Geometry and Object Modeling, Image processing such as Segmentation and Steganography , Speech signal processing, Artificial intelligent and applications, Computer Graphics and Data Security.