Generating Sunflower Random Polygons on a Set of Vertices

L.Mohammadi[2], A. Nourollah[1]

[1]Department of Electrical & Computer Engineering of Islamic Azad University, Qazvin, Iran

[2] Faculty of Electrical & Computer Engineering of Islamic Azad University, Qazvin, Iran

& Faculty of Electrical & Computer Engineering of Shahid Rajaee Teacher Training University, Tehran, Iran

Abstract - *Generating random polygons problem is important for verification of geometric algorithms. Moreover, this problem has applications in computing and verification of time complexity for computational geometry algorithms such as Art Gallery. Since it is often not possible to get real data, a set of random data is a good alternative. In this paper, a heuristic algorithm is proposed for generating sunflower random polygons using* $O(n \log n)$ *time.*

Keywords: convex hull, sunflower random polygon, visibility

1 Introduction

Computational geometry is a very important research field in computer science in which most computations are performed on known geometrical objects as polygons. Polygons are a convenient representation for many real-world objects; convenient both in that an abstract polygon is often an accurate model of real objects and in that it is easily manipulated computationally. Examples of their applications include representing shapes of individual letters for automatic character recognition, of an obstacle to be avoided in a robot's environment, or a piece of a solid object to be displayed on a graphic screen [7].

The generation of random geometrical objects has received some attention by researchers [2][3][6]. A challenge of these problems is the generation of random simple polygons. Since no polynomial time algorithm is known to solve the problem, researchers either try to use heuristic algorithms which don't have uniformed distribution or restrict the problem to certain classes of polygon such as monotone and star-shaped polygons [1][3][4].

The importance of geometric objects application is the simplicity of testing geometric algorithms. Since a set of data may become both too large and too hard to define for practical purposes, what one might do is to use randomly generated data that has a high probability to cover all the different classes of inputs. Thus, since practical data may not be available for testing, it is natural to test the algorithm on randomly input data.

Polygons are one of the fundamental building blocks in geometric modeling and they are used to present a wide variety of shapes and figures in computer graphics, vision, pattern recognition, robotics and other computational fields. Some recent applications address uniformed random generation of simple polygons with given vertices, in the sense that a polygon will be generated with probability $\frac{1}{T}$ if there exist a total of T simple polygons with such vertices.

One of the important geometry problems in which polygons play an important rule, is art gallery whose purpose is guarding a polygonal art gallery with the least number of guards (cameras). A well-known kind of art gallery problem is sunflower art gallery. The proposed question is this: What is the smallest number of guards required to protect the Sunflower Art Gallery?

Figur 1 shows a sunflower art gallery which is protected by 4 stationary guards. Some of the guards can not see through walls around corners of art gallery. Every point is visible at least one guard and it would be more economical to protect the gallery with fewer guards, if possible [5].

In this paper a heuristic algorithm is proposed for the generation of random sunflower polygons to estimate such problems.

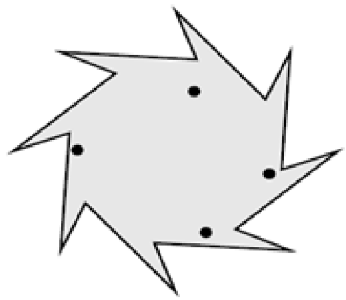

Fig. 1. The sunflower art gallery [5]

The following sections of this paper have been organized in this way: section 2 has been allocated to related works. In section 3 the needed preliminary concepts are stated. In section 4 the proposed algorithm is posed for generating random sunflower polygon and its performance and accuracy are investigated and finally in section 6 the conclusion will be discussed.

2 The related woks

Recently, the generation of random geometric objects and specially simple polygons has received some attention by researchers. For example, Epstein studied the uniformly random generation of triangulations [8]. Zhu et al. presented on an algorithm for generating x-monotone polygons on a given set of vertices uniformly at random [3]. A heuristic for the generation of simple polygons was investigated by O'Rourke and Virmani [6]. Auer and Held presented the following heuristic algorithms:

- *Steady Growth*: an incremental algorithm adding one point after the other whose time complexity in the worst case and in the best case is $O(n^2)$ and $O(n\log n)$, respectively.
- *Space Partitioning*: which is a divide and conquer algorithm and it has time complexity of $O(n^2)$.
- *Permute & Reject*: which creates random permutations (polygons) and surveys whether it is corresponding with a simple polygon or not, until a simple polygon is encountered. Its complexity is $O(n\log n)$.
- *2-opt Moves*: which by starting from a completely random polygon and replacing its intersected edges encounters a simple polygon and its complexity is $O(n^4)$ [2].

3 Preliminaries

Let S be a set of random vertices, there exists T simple polygons on S in total, such that every polygon is generated with probability $\frac{1}{\mathrm{T}}$. it is supposed that no three points are linear. A simple polygon, is a limited plane by a limited set of line segments that form a simple closed curve. In other words, a simple polygon P on S, is a polygon whose edges don't intersect one another except on vertices S .

A convex polygon, is a simple polygon which for both of vertices x and y from polygon, the line segment $\overline{xy}$ lies inside P or on its border, i.e., $\overline{xy} \subseteq P$. The convex hull of a finite set of points on the plane ($CH(S)$) is the smallest convex polygon P that encloses S. The smallest polygon means that there is no polygon $\acute{P}$ such that $S \subseteq \acute{P} \subset P$ (Figure2).

Let k be the number of all convex hull layers on the set of points S. Every layer is defined by l_{c} ($k > 1$ and $1 \le l_{\mathrm{c}} \le k$). By the supposition of numbering layers from the most internal to the most external convex hull layers, n_{i} is supposed to be the number of the most internal convex hull layer, i.e., l_1 .

Vertex x sees vertex y if $\overline{xy} \subseteq P$ and $\overline{xy}$ doesn't lie out of P. In this case y is visible from x or in other words x has visibility toward y. The polar sorting, is the sorting of a set of vertices around a given vertex according to polar angle. The star-shaped polygon, is a polygon which is visible at least from an interior vertex. In the next section, the proposed heuristic algorithm is posed to generate random sunflower polygon.

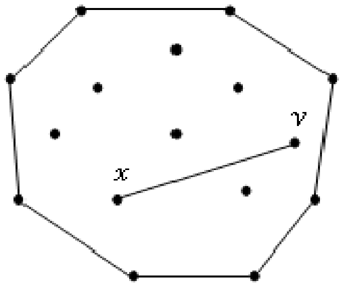

Fig. 2. The convex hull of set of points S

4 The proposed algorithm

In this section, an algorithm is posed to generate the random sunflower polygon with time complexity $O(n\log n)$ in which n is the number of vertices.

To generate a random sunflower polygon on a vertex set it is performed in this way that first, According to Graham greedy algorithm, the assumption $CH(S)$ is obtained. The Graham algorithm for the generation of $CH(S)$ performs in this way:

- First, the vertex with lowest y-coordinate which is the rightmost point, has selected from the point set S and is called p_0.
- All the remaining points are sorted around p_0 according to polar angle $(p_1, \dots, p_{n-1})$.
- A stack is constructed and p_0 and p_1 are pushed to it.
- By starting from point p_2 to $p_{\mathrm{n-1}}$ for every vertices the following case is investigated:
- p_{i} is pushed in the stack if it is left of two tops of the stack and is incremented counter i; otherwise, the top of stack is removed if p_{i} is right of two tops of the stack (Figure 3).

Algorithm: Graham Scan
Find rightmost lowest point; lable it p_0.
Sort all other points angulary about p_0.
Stack $S = (p_1, p_0) = (p_t, p_{t-1}); t$ indexes top.
$i = 2$
While $i < n$ **do**
  If p_{i} strictly left of $p_{t-1}p_t$ **Then**
    Push(p_i, S) and set $i \leftarrow i + 1$
  Else
    Pop(S)
  End If
End While

Fig. 3. The Graham scan algorithm

Time complexity of Graham algorithm is $O(n\log n)$. After determining $CH(S)$, while n is opposite of zero, Graham algorithm is recalled for any remaining point set S. Let the number of points on the most internal assumption convex hull, i.e., l_1 be n_{i}, the point set S can be partitioned to n_{i} sections. This partitioning is in this way that by starting from the leftmost point on l_1 and in counter-clockwise direction, every edge l_1 is continued from the second point. Thus, n_{i} set of points are obtained (Figure 4).

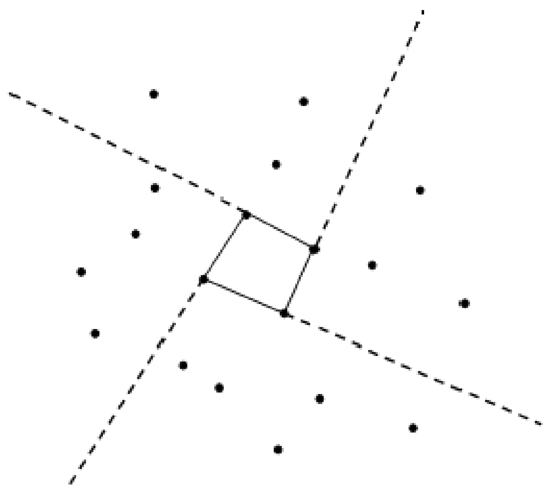

Fig. 4. Partitioning of the set if points S by continuing the second vertex of every edge l_1

After partitioning of the point set S, by starting from the leftmost point or the vertex with least x-coordinates on l_1 on the first part of partitioned vertices, the points of the part until to starting point of the next part, are connected together in this way: points are sorted around point v_{j} on l_1 $(1 \leq j \leq n_i)$ according to polar angle in counter- clockwise direction. Then these points sorted in order are connected together from v_{j} to $v_{\mathrm{j+1}}$.

Thus, by scanning of all parts and connecting points of every partitioned part based on explained process, all points of set S are connected together and a simple polygon P is resulted which is not necessarily a star-shaped polygon (Figure 5). Procedure of the algorithm is following way:

- while, n is the opposite of zero, all assumption convex hull layers are computed on the set of points S based on Graham algorithm.
- The number of points of assumption convex hull l_1 are counted and are supposed as n_{i}.
- Every assumption edge $v_{\mathrm{j}}v_{\mathrm{j+1}}$ on l_1 is continued from vertex $v_{\mathrm{j+1}}$ and the set of points are partitioned to n_{i} sets.
- For every resulted point set, it is performed in this way: all point of that part have been sorted around v_{j} based on polar angle and then points of between v_{j} and $v_{\mathrm{j+1}}$ sorted in order are connected together from v_{j} to $v_{\mathrm{j+1}}$.
- The resulted random simple polygon, is a sunflower polygon. In the special case which $n_{\mathrm{i}} = 1$, it is called star-shaped polygon.

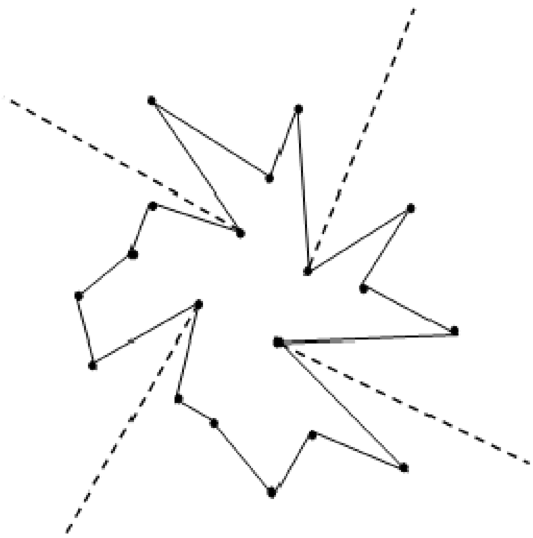

Fig. 5. Generation of a sunflower polygon with connecting the sorted points of any partition together

4.1 Verification of algorithm performance

In this section, the performance of proposed algorithm is investigated by computing its time complexity. In this algorithm, time of obtaining the most external convex hull layer is $O(n \log n)$ based on Graham algorithm which it would be the maximum time needed to generate of convex hull since, it is computed on n points. Partitioning of points S to n_{i} sets, requires $O(n_i)$ time that is in the best case $O(1)$, i.e., in the case of existing only a resulted part of points partitioning. Finding the leftmost point with least x-coordinates on l_1 with the number of n_{i} points, is $O(n_i)$. Sorting points of the any part set, in the case the sprawl of points is good, i.e., $\frac{n}{n_{\mathrm{i}}}$ points exist in any part averagely, is $O(\frac{n}{n_{\mathrm{i}}} \log \frac{n}{n_{\mathrm{i}}})$. In the worst case, which exists only one partitioned part, this sorting takes $O(n \log n)$ time. Therefore, time complexity is $O(n \log n)$ in total.

4.2 Determining of visible polygon

In this section, the posed heuristic algorithm is investigated. This algorithm performs properly for any set of given points with the number of assumption convex hull layers $k > 1$ in general case.

To proof the accuracy of this algorithm, it is discuss every set of partitioned points and the visibility of all points in every part from one point on l_1. Since, for generating this random polygon, all partitioned parts are independent of one another and they haven't subscription together and also as in every part, all its point set are visible from v_j on l_1, because of polar sorting of points around v_j, it is verified simply that a random sunflower polygon is inevitably generative by this proposed algorithm on any set of given points with number of assumption convex hull layers $k > 1$.

5 Conclusions

In this paper, a heuristic algorithm with time complexity of $O(n \log n)$ was proposed for the generation of random sunflower polygon. This algorithm can be used to estimate many algorithms and geometric problems such as art gallery. Also it was proved that this algorithm has properly on any set of given points with intricate assumption convex hulls (with the number of layers $k > 1$) and it generates a random sunflower polygon.

6 References

[1] McDonald, B., Smith, I. “Partial symmetry breaking”; In: P. Van Hentenryck(Ed.), Proc. Of CP'02 LNCS2470, Springer-verlag, 431—445, 2002.

[2] Auer, T., Held, M. “Heuristics for the Generation of Random Polygons”; Proc. 8th Canadian on Computational Geometry(CCCG'96), 38—41, 1996.

[3] Zhu, C., Sundaram, G., Soneyink, J., Mitchell, J.S.B. “Generating Random Polygon with Given Vertices”; Comput. Geom. Theory and Appl., Vol 6, Issues 5, 277—290, 1996.

[4] Sohler, C. “Generating random star-shaped polygons”; In: Proc. 11th Canadian Conference on Computational Geometry(CCCG'99), 174—177, 1999.

[5] Epstein, P. “Generating Geometric Objects at Random”; Master's thesis, CS Dept., Carleton University Ottawa K1S5B6, Canada, 1992.

[6] O'rourke, J., Virmani, M. “Generating Random Polygons”; Technical Report 011, CS Dept. Smith Colege, Northhampton, MA 01063 1991.

[7] Berg, M.D. “Computational Geometry Algorithms and Applications”; 3rd , Published by springer-verlag, 2008.

[8] Michael, T.S. “How to Guard an Art Gallery and other Discerete Mathematical Adventures”; the Book of the Johns Hobkins University Press, printed in the United Stetes of American, 2009.