Three Heuristic Algorithms for Generation of Random Polygons by a Given Set of Vertices

A. Nourollah[1], **L.Mohammadi**[2]
[1] Faculty of Electrical & Computer Engineering of Islamic Azad University, Qazvin, Iran
& Faculty of Electrical & Computer Engineering of Shahid Rajaee Teacher Training University, Tehran, Iran
[2] Department of Electrical & Computer Engineering of Islamic Azad University, Qazvin, Iran

Abstract - *One of the discussed issues in computer graphics and computational geometry is the generation of random polygons. This problem is of considerable importance in the practical evaluation of algorithms that operate on polygons, where it is necessary to check the correctness and to determine the actual CPU-usage of an algorithm experimentally. To this time, no polynomial-time algorithm is known for the random generation of polygons with a uniform distribution. In This paper, three new inventive algorithms are presented for generating random polygons using a set of random vertices using* $O(n\ logn\ )$ *time.*

Keywords: angular scanning, convex hull, random Simple polygon, star-shaped polygon, computer graphics.

1 Introduction

The computer graphics is the science of computing the images which would be led to generate them. One of the important branches of computer graphics is the computational geometry. The computational geometry executes some computing on the geometrical shapes such as polygons. The polygons are appropriate shapes for representing the shapes of real world and every object in the nature would be representable as a set of polygons. For the reason of this issue importance, one of the important discussed issues in computer graphics and computational geometry is the problem of generating random polygons.

The problem of random polygon generation is defined as follows: Given S, a uniformly random polygon on S is a polygon generated with probability $\frac{1}{k}$ if there exist k simple polygon on S in total.

Consider two diffrenet pairs of points the two edges defined by these pairs. It is easy to observe that the number of simple polygons containing one of these edges by a dashed line: For the first edge there exactly two simple polygons (Fig 1-a, Fig 1-b) which contain it,whereas for the second edge three simple polygons (Fig 1-c, Fig 1-d, Fig 1-e) containing it exist. Unfortunately no algorithm has been presented yet for the problem of generating random simple polygons with linear time complexity. This motivated researchers to pursue heuristics for generating random simple polygons.

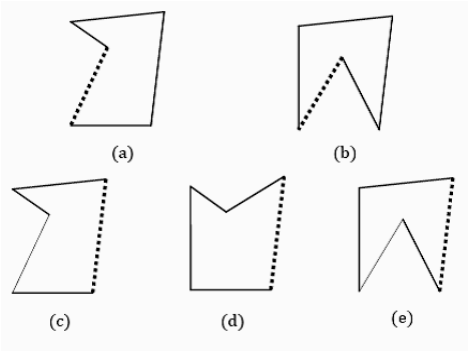


Fig. 1. Two edges belonging to a different number of simple polygons

The generation of random simple polygons has two main areas of application: The empirical verification of an algorithm oprating on polygons. The practical testing of its CPU-usage. Too many algorithms have been presented for solving the problem of random polygons and some of them are briefly explained here in this paper [1].

The following sections of this paper has been organized in this way: section 2 has been allocated to related works for generating of random polygons. In section 3, three proposed algorithms would be offered and section 4 investigates the performance of the discussed algorithms and finally in section 5 conclusions would be discussed.

2 The related work

For the reason mentioned in section ago, the random generation of geometric objects has received some attention by researchers: Epstein and Sack presented an $O(n^4)$ algorithm for generating of simple polygons at random [2].

Devroy, Epstein and Sack studied the random generation of intervals and hyperrectangles. They consider the problem of generating a random hyperrectangle in a unit hypercube [3].

Atkinson and Sack studied the uniform generation of forests of restricted height. A k-way tree is either empty or consists of a root node and an ordered sequence of k subtrees. their

algorithms runs in $O(n^3.h)$, where h denotes the height of the tree[4].

An algorithm running in at most $O(n^2)$ time for the random generation of x-monotone polygons was described by Zhu et al [5]. An interesting approach for the generation of random polygons in the plane (but not on a given set of points) was researched by O'Rourke and Virmany[6]. In the next section three proposed heuristic algorithms are designed for this problem.

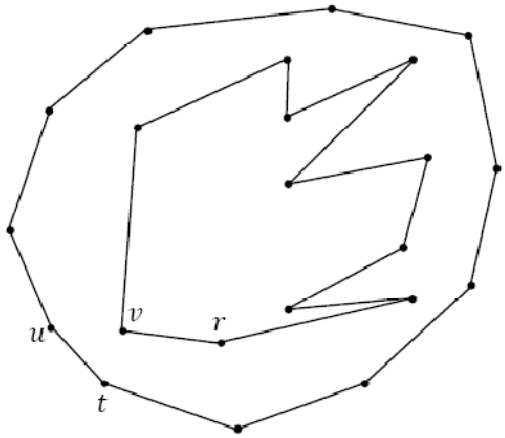
3 Proposed algorithms

In this section, three proposed algorithms are discussed for generating of random polygons. each algorithm has time complexity $O(\,n\,logn\,)$.

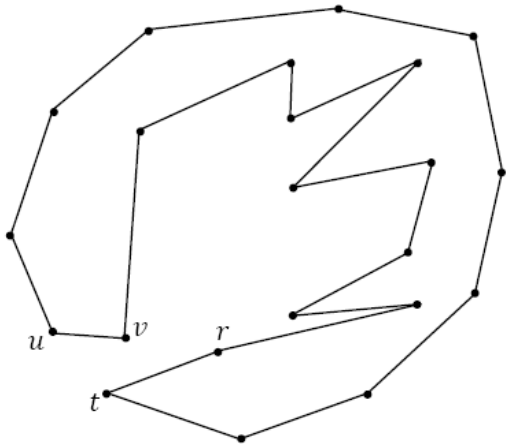
3.1 Algorithm 1

Let S be a set of random vertices and no three points are on the same line. The convex hull, the set of S vertices in the plane ($CH(S)$), of a point set S in the plane is the enclosing convex polygon with smallest area. With this description, this algorithm performs in a way that first $CH(S)$ would be obtained. Let $\acute{S}$ be a set of interior vertices of this convex hull. Then, for the set of those vertices which are inside it will perform in this way, i.e., : first the vertex with the least y-coordinate is found. It is supposed that this vertex is v. The vertices of $\acute{S}$ is sorted according to the polar angle in the counter-clockwise around v. By angular scanning of these vertices around vertex v, visited vertices would be joined together. The last visited vertex is joined to vertex v and in this way a star-shaped polygon would be obtained (that from a point inside it all vertices are visited). Now, from the set of vertices over $CH(S\,)$, the closest vertex to v is selected and it is supposed to be this vertex u. If the vertex close to v over interior star-shaped polygon in the counter-clockwise is r and the vertex close to vertex u over $CH(S)$ in the counter-clockwise is t (Fig 2-a), so the edges $(v,r\,)\,,(u\,,t\,)$ with the edges $(v,u\,)\,,(t\,,r\,)$ are replaced. Thus, a random polygon would be created (Fig 2-b). The procedure of this algorithm is as the following:

- First, $CH\,(S)$ would be computed over the set of random vertices S.
- The set of interior vertices $CH(S)$ is considered as $\acute{S}$ and the vertex with the least y-coordinate is selected from the set of vertices (vertex v).
- The vertices $\acute{S}$ are visited according to the polar angle in the counter-clockwise and a star-shaped polygon is obtained with angular scanning of vertices around vertex v and joining the sorted vertices.
- The closest vertex to vertex v is selected From the set of vertices over $CH(S\,)$ (vertex u).
- The vertex close to v and the vertex close to u in the counter-clockwise is called r and t, respectively.
- The edges $(v,r\,)\,,(u\,,t\,)$ are replaced by the edges $(v,u\,)\,,(t\,,r\,)$.



(a)



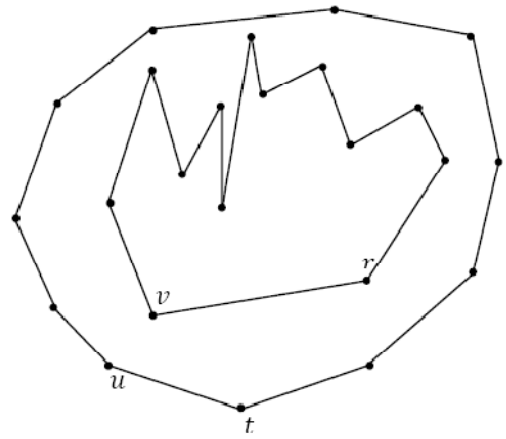
(b)

Fig. 2. (a) Creating a star-shaped polygon with the points inside the convex hull, (b) creating the random polygon with the replacement of edges $(v,r\,)\,,(u\,,t\,)$ with the edges $(v,u\,)\,,(t\,,r\,)$.

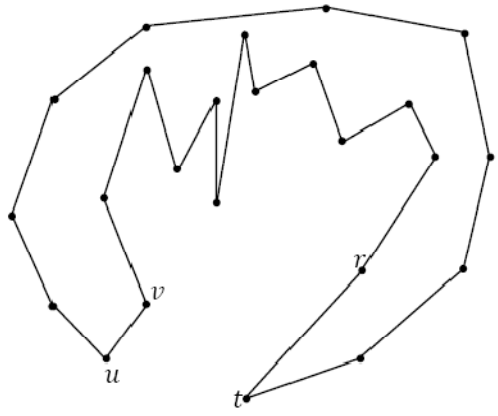
3.2 Algorithm 2

In this algorithm like the previous algorithm, first $CH(S\,)$ is computed. Let $\acute{S}$ be the vertex set inside of the convex hull. For this set of vertices, the existing convex bottom algorithm would be applied which has $O(\,n\,logn\,)$ time and performs in this way that first two vertices which has the least and the most x-coordinate is considered. These vertices is joined together using a presumptive line so that the set of vertices would be divided into two upper and lower half (if the set of vertices in the lower half is empty, the presumptive line is considered as the convex hull of two points and the algorithm executes again).

Now, the convex hull of the presumptive line's lower half is computed and the presumptive line is deleted. All remaining vertices would be sorted from left to right in the order of x and the most left and the most right points would be joined to the left and right end vertices of convex hull, respectively. Then, like the above algorithm, vertices v, u, r and t have been considered as it's explain before in the previous algorithm (Fig 3-a) and the edges $(v, r), (u, t)$ is replaced with the edges $(v, u), (t, r)$ (Fig 3-b). the created polygon is a simple random polygon.



(a)



(b)

Fig. 3. (a) Creating a polygon with the Convex Buttom algorithm, (b) creating the random polygon by replacing the of edges $(v, r), (u, t)$ with the edges $(v, u), (t, r)$.

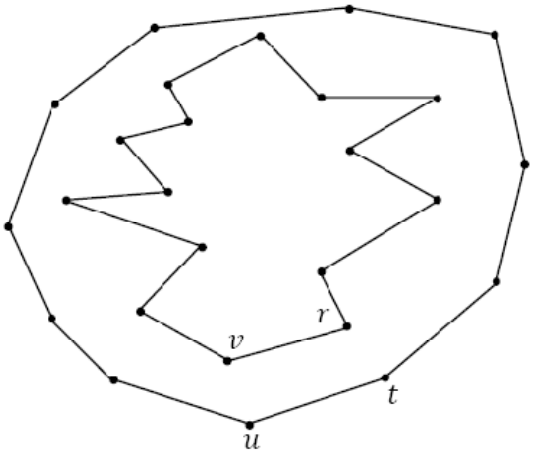
The procedure of this algorithm with assumption the existing convex bottom algorithm is as the following:

- First, $CH(S)$ would be computed over the set of random vertices S.
- The set of interior vertices $CH(S)$ is considered as $\acute{S}$ and the vertex with the least y-coordinate is selected from the set of vertices (vertex v).
- The Convex Bottom algorithm is called for those vertexes which are inside of the convex hull (the set $\acute{S}$).
- The closest vertex to vertex v is selected From the set of vertices over $CH(S)$ (vertex u).
- The vertex close to v and the vertex close to u in the counter-clockwise is called r and t, respectively.
- The edges $(v, r), (u, t)$ is replaced with the edges $(v, u), (t, r)$.

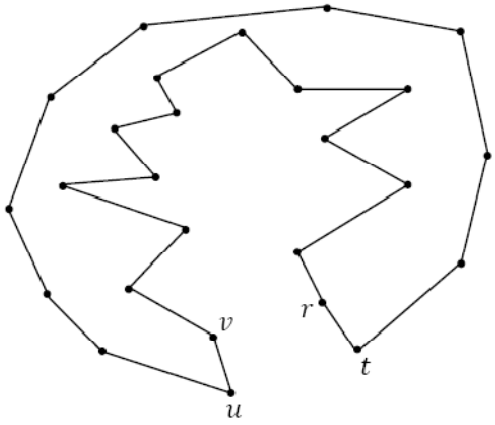
3.3 Algorithm 3

In this algorithm like the two previous algorithms, first $CH(S)$ is computed. Let $\acute{S}$ be the vertex set inside of the convex hull $\acute{S}$. for this set of vertices, the existing TwoPeasants algorithm (for the presumptive line according to y-coordinate) would be applied which it has time complexity $O(n\,logn)$ and performs in this way that first two vertices which has the least and the most x-coordinate is considered. These vertices would be joined together using a presumptive line so that the set of vertices would be divided into two upper and lower half (if the set of vertices is empty, the algorithm executes again) [7]. The next steps are as the following: The upper half vertices would be sorted while they started from the left end point and the lower vertices would be sorted like this way and the end vertices would be joined from both sides and like two previous algorithms the vertices v, u, r and t as it's explained before in section 3-2 have been considered (Fig 4-a) and the edges $(v, r), (u, t)$ is replaced with the edges $(v, u), (t, r)$. The shape which is obtained is a simple random polygon (Fig 4-b). Thus the created polygon in this way is a simple random polygon. The procedure of this algorithm with assumption the existing TwoPeasants algorithm is as the following:

- First, $CH(S)$ would be computed over the set of random vertices S.
- The set of interior vertices $CH(S)$ is considered as $\acute{S}$ and the vertex with the least y-coordinate is selected from the set of vertices (vertex v).
- The TwoPeasants algorithm is called for those vertexes which are inside of the convex hull (the set $\acute{S}$).
- The closest vertex to vertex v is selected From the set of vertices over $CH(S)$ (vertex u).
- From the set of vertices over $CH(S)$, the closest vertex to vertex v is selected (vertex u).
- The vertex close to v and the vertex close to u in the counter clockwise is called r and t respectively.
- The edges $(v, r), (u, t)$ is replaced with the edges $(v, u), (t, r)$.



(a)



(b)

Fig. 4. (a) Creating a polygon with the TwoPeasants algorithm, (b) creating the random polygon by replacing the edges (v, r), (u, t) with the edges (v, u), (t, r).

In the next section, the performance of proposed discussed algorithms would be evaluated.

4 Performance evaluation

In this section, performance and time complexity of the proposed algorithms would be investigated. in the proposed algorithm 1, since the time complexity the computation of $CH(S)$, is $O(n\,logn)$, finding a vertex with the least y-coordinate (vertex v), is $O(n)$, and finding the closest vertex to vertex v from the vertices set $CH(S)$ is $O(n)$, Likewise, since the time complexity of sorting of the set of vertices S according to the polar angle around vertex v is $O(nlogn)$ and the replacement the edges (v, r), (u, t) with the edges (v, u), (t, r), would be performed in linear time, therefore the time complexity for this algorithm is $O(n\,logn)$.

For the proposed algorithm 2, since time order for computation $CH(S)$, is $O(n\,logn)$, and finding a vertex with the least y-coordinate (vertex v), is $O(n)$, and the time complexity of convex bottom algorithm is $O(n\,logn)$, so time complexity for this algorithm is $O(n\,logn)$, as well.

In the proposed algorithm 3, time order for computation $CH(S)$, is $O(n\,logn)$, finding a vertex with the least y-coordinate (vertex v) is $O(n)$ and time complexity of Towpeasants algorithm is $O(n\,logn)$. Thus, similar to previous algorithms, time complexity of this algorithm is $O(n\,logn)$. In the next section, the conclusions would be investigated.

5 Conclusions

In this paper, a review of existing algorithms was done for generating random polygons. Many algorithms have been executed for generating random polygons but no algorithm has been presented yet for the problem of generating simple random polygons in linear time. This problem causes to apply heuristic approaches. In this paper, three inventive algorithms have been proposed for generating simple random polygons which has time complexity of $O(n\,logn)$. Since the generation of simple polygon is not possible in less than $O(n\,logn)$ time, so these algorithms have optimal order.

6 References

[1] Auer, T. “Heuristic for generation of polygons”; In: Proc. Canada. Conf. Compute. Geom , 34--44, 1994.

[2] Epstein, P., Sack, J.-R. “Generating objects at Random”; Master's thesis CS Dept., Carleton university, Ottawa K1S 5B6, Canada, 1992.

[3] Derroy, L., Epstein, P., Sack J.-R. “on generating random intervals and hyperrectangles”; J.Compute. Graph.Stat , 291--307, 1993.

[4] Atkinson, M.D., Sack, J.-R. “Uniform generation of forests of restricted height”; Inform. Process. Lett, 323—327, 1994.

[5] Zhu, C., Sundaram, G., Snoeyink, J., Mitchell, J.S.B. “Generating random polygon with given vertices”; Compute. Geom.: Theory Applic, 1996.

[6] O'Rourke, J., Virmani, M. “generating random polygons”; Technical report 11. CS Dept. Smith College, MA, USA , 1991.

[7] http://www.geometrylab.de/polygon/Random Polygon.html.en