Art Gallery Problem

Abhishek Kaswan
Information Technology Department
National Institute of Technology
Karnataka
Surathkal, India
myritu06@gmail.com

Jaidev Chittoria
Information Technology Department
National Institute of Technology
Karnataka
Surathkal, India
jaidev.chittoria02@gmail.com

Ayush Bhandari
Information Technology Department
National Institute of Technology
Karnataka
Surathkal, India
ayushbhandari8909@gmail.com

Abstract— The Art Gallery Problem (AGP) is one of the classic problems in Computational Geometry.Many variants of these problems have already been studied.In this paper, we propose an algorithm to solve the art gallery problem in which guards are placed on the vertices of the polygon P i.e gallery.
.

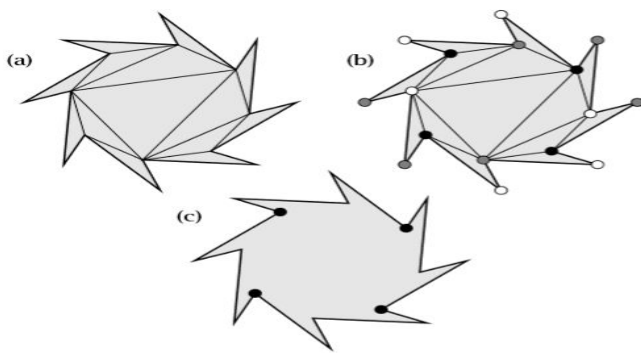
I. INTRODUCTION

The problem was first posed to Vaclav Chv´atal by Victor Klee in 1973 and was stated as: Consider an art gallery, what is the minimum number of stationary guards needed to protect the room? In geometric terms, the problem was stated as: given a n-vertex simple polygon, what is the minimum number of guards to see every point of the interior of the polygon? art gallery questions involve issues in computational geometry, a large and active field that blends geometry with ideas from discrete mathematics and optimization . Chv´atal was able to prove that for simple polygons floor of n/3 guards is sufficient to guard an art gallery and sometimes necessary, when there are n vertices in the polygon . He basically gave an upper bound for minimal no. of guard.

<u>Art gallery theorem:</u>

g(n)=n / 3 for n =3, 4, 5,....

In 1978 Steve Fisk constructed a much simpler proof via triangulation, which is a method of decomposing a polygon into triangles, and coloring of vertices.



First, partition the gallery into triangles by inserting suitable noncrossing diagonals, as in (a). Then assign one of the colors from blue, red, or green to each of the n corners so that every triangle has one corner of each color. In (b) we have n = 16, and there are four bluek, six black, and six green corners. Finally, if we post guards at the four blue corners, then every triangle is certainly protected (since every triangle has a blue corner), and hence the entire gallery is protected by the guards in (c).

The many real life applications of AGP not only inspired the mathematics community ,but also inspired many variations of the problem that modeled real-life situations.One of the many extensions of this problem is the chromatic art gallery problem, which aims to determine the minimum number of colors required to color a guard set, a set of vertices in an n-vertex polygon. A guard set is colored such that no two conflicting guards have the same color, where two conflicting guards are those whose areas of visibility overlap.
Input: art gallery G(n) with n walls

Output: positions for at most n/3 guards that protect G(n)

1. Triangulate G(n) by inserting suitable diagonals. 2. Find a polychromatic 3-coloring of the corners of the triangulation. 3. Post guards at the vertices with the least used color.

A. PROBLEM STATEMENT

Guarding an art gallery with the minimum number of guards who can keep a check on the whole gallery.

We have considered the layout of the art gallery be a simple polygon in which guards are to be placed on the vertices of the polygon . Let a set S of points is said to guard a polygon if, for every point p in the polygon, there is some q ∈ S such that the line joining p and q does not leave the polygon.

B. OBJECTIVES

The objective of this project is to first Model the art gallery as a region bounded by some simple polygon (no self-crossing and no holes) then Triangulate it with a fast algorithm then we will generate a 3-coloring by DFS (as presented earlier).Then we will take the smallest color class to place the cameras.Display the visibility of each guard corresponding to solution.

We represented gallery by a simple polygon. Guards or Cameras are assumed to have a viewport of 360 degrees. And they can see as far as there is no obstacle in the path or a wall. For now We would ignore the size of the Guards(cameras). Guards will be represented by points in the polygon. The Guard(camera) sees all points it can be connected to by a segment lying entirely inside the polygon.

II. LITERATURE SURVEY

The following papers were an inspiration to this project, and have therefore been listed here:

1. An Exact and Efficient Algorithm for the Orthogonal Art Gallery Problem

Authors- Marcelo C. Couto ; Cid C. de Souza ; Pedro J. de Rezende

Work Done-
Proposed an exact algorithm to solve the orthogonal art gallery problem in which guards can only be placed on the vertices of the polygon P representing the gallery. Their approach is based on a discretization of P into a finite set of points in its interior.

Advantages-
Provides clear idea about implementation.[n/4] guards are sufficient where n is the number of vertices.

Disadvantages-
Can become complicated.

2. Swarm of agents for guarding an Art Gallery: A computational study

Authors- Mahdi Moeini ; Daniel Schermer ; Oliver Wendt

Work Done-
Proposed two algorithms with visualisation for a version of AGP where guards are autonomous and have lack of communication abilities.For this purpose, a self-contained simulator was designed which was able to read or generate a non convex polygon and also to simulate the movements of guards inside the polygon was done by using navigation algorithms.

Advantages-
This project uses two algorithms and each algorithm has a better performance on specific types of polygons.

Disadvantages-
It is complex as in this version of AGP guards are autonomous and have limited communication abilities.

3.The Quest for Optimal Solutions for the Art Gallery Problem: A Practical Iterative Algorithm

Authors- Davi C. TozoniPedro J. de RezendeCid C. de Souza

Work Done-
Proposed a solution for the Art Gallery Problem which is a practical iterative algorithm with point guards, in this approach it finds a decreasing upper bounds and increasing lower bounds until optimal value is achieved.

Advantages-The algorithm gives optimal results on a very large collection of instances or variety of without holes-polygon having vertices ranging from 20-1000.

Disadvantages- Not works for all variety of without holes-polygons.

4.Approximation algorithms for art gallery problems in polygons

Author- Subir Kumar Ghosh

Work Done-
This paper proposed approximation algorithms for minimum vertex and edge guard problems for polygons with or without holes with a total of n vertices.

Advantages-For simple polygons, approximation algorithms for both of the problems run in O(n^4) time and gives solution which can be at most O(logn) times of the optimal solution.

Disadvantages-Algorithm takes O(n^4) running time.

III. IMPLEMENTATION

first we build a polygon and used a graphical user interface to show instruction that how we can use the tool. In design part we also used prev and next buttons where user can navigate and visualize the implementation.

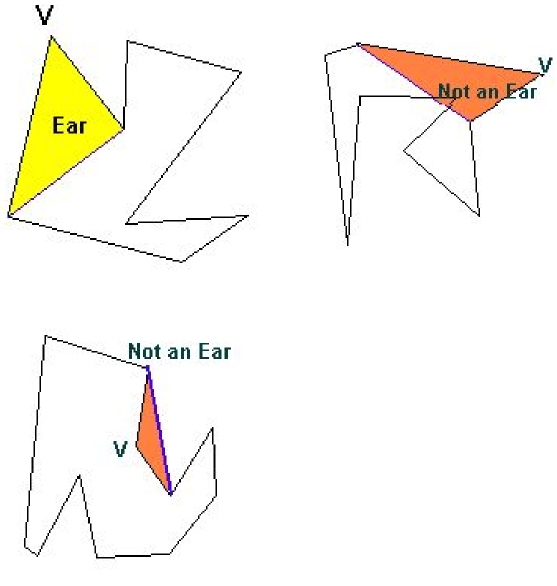
We have used the following algorithms in this tool:

- Ear Clipping Algorithm for Triangulation
- M-coloring using backtracking for 3-coloring
- Polygon Visibility

Brief discussion of these algorithms are as below:

1.Ear Clipping Algorithm for Triangulation:

The ear clipping triangulation algorithm consists of searching an ear and then cut- ting it off from current polygon. The original version of Meisters's ear clipping algorithm runs in O(n3) time, with O(n) time spent on checking whether the formed triangle is valid.

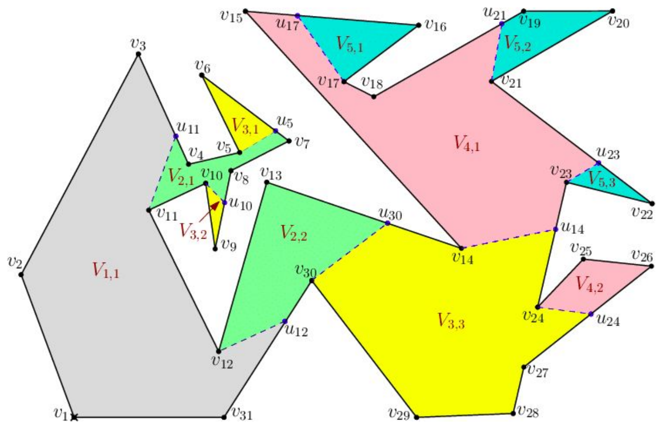


2. 3-coloring using backtracking:

In this problem, an undirected graph is given. There is also provided 3 colors. The problem is to find if it is possible to assign nodes with 3 different colors, such that no two adjacent vertices of the graph are of the same colors. If the solution exists, then display color corresponding to the vertex.

Starting from vertex where creation of polygon started, we will try to assign colors one by one to different nodes. But before assigning, we have to check whether the color of adjacent vertices are different or not..

3.Polygon Visibility:



In this problem simple polygon is given and we move along the edges with starting a vertex and if it is visible and it has no invisible edges between previous visible vertex and v then add v to solution .Otherwise move to closest intersection point of the invisible edge and keeps doing it until it goes through the whole polygon.

IV. DESIGN AND METHODOLOGY

1).Creation of a polygon in GUI

2.)Triangulation using Ear Clipping

Algorithm:

i.Given initial list of vertices V2.

ii.Construct initial list R of reflex vertices and construct list E of ear tips using list

iii.Start with point(vertex) of creation of polygon and remove one ear tip at a time i.e V i from E

->Add triangle Vi-1ViVi+1to final triangulation

->Remove Vi from list V

->Update R and E with adjacent vertices Vi-1, Vi+1

->If the adjacent vertex is reflex, then there is possibility that it becomes convex and possibly, an ear

->If an adjacent vertex is an ear,then it does not necessarily remain an ear

iv. Repeat 3. until list V contains only 3 vertices –last triangle of triangulation

3.) 3-coloring algorithm

i) isvalid takes Vertex, colorList to check, and color, which is trying to assign checks if the color assigning is valid, otherwise false.

```
for all vertices v of the graph
        if there is an edge between v and i
         and col = colorList[i],    then
                 return false
return true
```

ii)graphColoring consists of the list containing vertices and corresponding color assigned to that vertex and the starting vertex checks colors are assigned or not.

```
Pick a random vertex
 i. if all vertices are checked, then
       return true
 for 3 colors col from available colors
       if isValid(vertex, color, col), then
           add col to the colorList for vertex
       if graphColoring(colors, colorList, vertex+1)
       =true,    then
              return true
 remove color for vertex

return false
```

4.)Polygon Visibility

Algorithm:

i. Given a simple polygon and a vertex

ii.Maintain a list for storing invisible_edges = []

iii.For every other vertex V:

-> 1.if V is visible:

A. if no invisible edges between previous visible vertex and V:

Add V to solution

B. else:

i) Find the closest intersection
point of the invisible edges
with line[vertex,previous visible V]
and line[vertex, V] from the vertex such that
the V or previous visible V is
on the segment[vertex, intersection

ii) if the 2 edges that meet at
previous visible V are on same sides
of line[vertex, previous visible V]:

-> if the line from closest
intersection point for
line[vertex, previous visible V]
to previous visible V is
in polygon:

Add closest intersection
point to solution

iii) if the 2 edges that meet at
previous visible V are on
same sides of line[vertex, V]

->if the line from closest
intersection point for
line[vertex, V] to V
is in polygon:

Add closest intersection
point to solution

Add V to solution

->2. else if V is not visible:

add V to invisible_edges

5).Display of minimum guards

V. SAMPLE INPUTS AND OUTPUTS

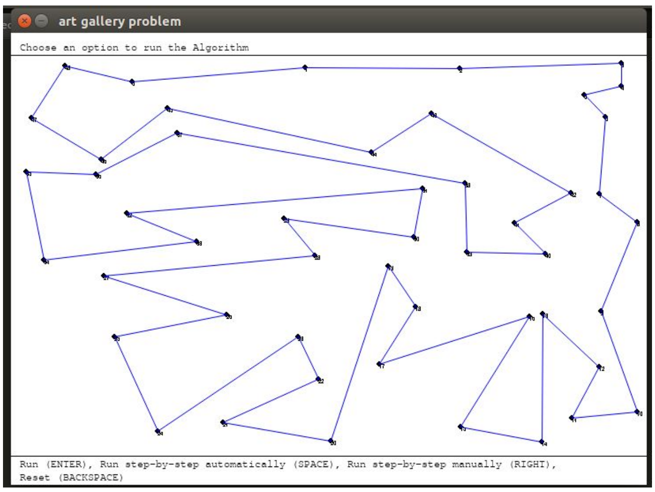


Fig 1.Creation of a polygon
n=48

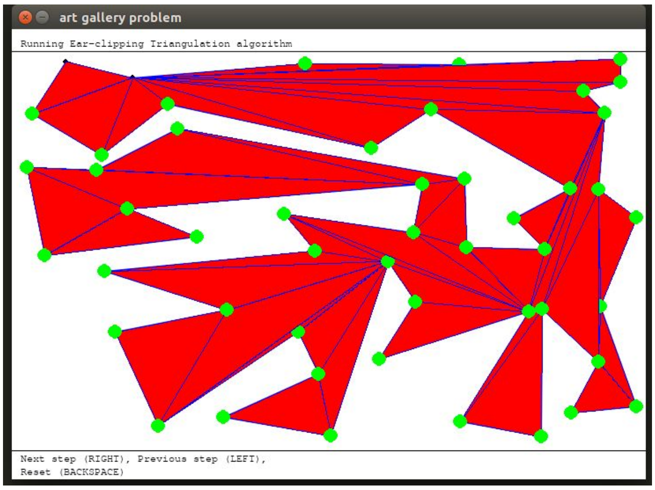


Fig 2.Triangulation

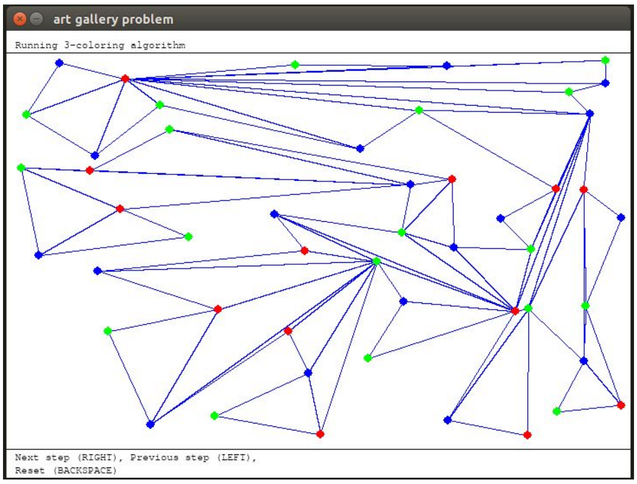


Fig 3. Three-coloring using DFS

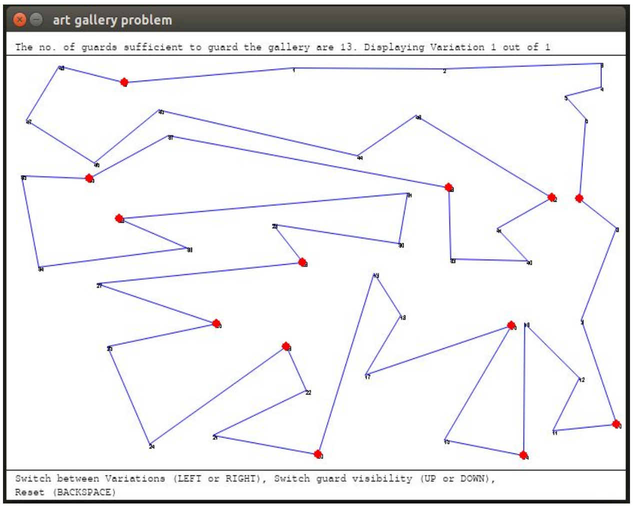


Fig 4. Display of result with the position of
guards in polygon
No.of guards sufficient to guard the gallery are 13.

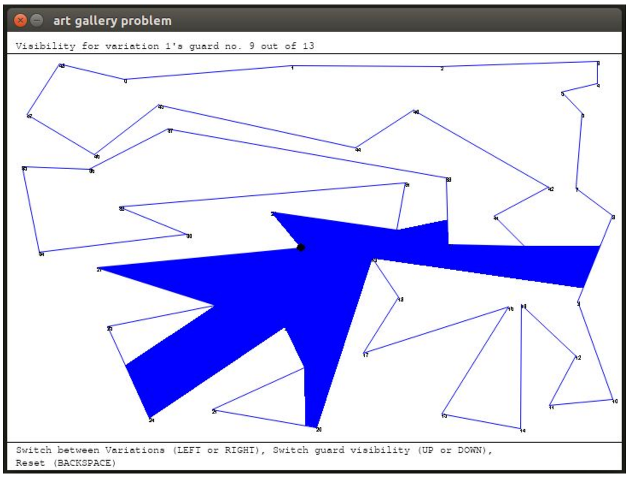


Fig 5. Visibility of a guard corresponding to the given solution

VI. CONCLUSION

Finding the exact number of point guards required to guard the entire room is an NP-Hard problem.Steve Fisk claims that to guard a simple polygonal room with n vertices, you never need more than floor(n/3) vertex guards. In this tool, we find the sufficient number of vertex guards (points placed on the vertices of the polygon) required to guard the polygonal room with n vertices in O(n log n) time using Fisk's proof.
these are some issues we are facing within the project:
• Polygon Visibility is inaccurate in rare cases because of floating point comparisons (issues with precision).
• Triangulation may fail sometimes if the points are placed too close to each other (bug in the ear clipping algorithm).
• Does not always give minimum no. of guards (flaw in the approach, finding accurate answer always is NP-hard problem).

VII. FUTURE WORK

We will try to Efficient Vertex guard visibility algorithm.This project isn't not ready to edit polygon during run time. so in the future we will improve the project and try to edit polygon during run time (by drag and drop of edges and vertices) and Ability to pause the visualization of the algorithm.

VIII. APPLICATIONS

Solutions to the Art Gallery problem have provided ideas for improving securities .For example, where, on college campuses, are the best locations to place security officers and how many are needed?
we can also use AGP in Placement of radio antennas,Architecture, Urban planning , Gaming and graphics,Ultrasonography and sensors.

IX. INDIVIDUAL CONTRIBUTION

- Abhishek Kaswan (18IT201) :User Interface, functionality of AGP.
- Ayush Bhandari (181IT209) :computing polygon visibility for solution vertices.
- Jaidev Chittoria(18IT119) :compute triangulation, 3 coloring implementation.

X. REFERENCES

1. O'Rourke, J. Art Gallery Theorems and Algorithms. New York: Oxford University Press.
2. DIMACS Research and Education Institute. "Art Gallery Problems."
3. Abrahamsen, Mikkel; Adamaszek, Anna; Miltzow, Tillmann (2017), The Art Gallery Problem is ∃ R-complete
4. Chvátal, V. "A Combinatorial Theorem in Plane Geometry." J. Combin. Th. 18, 39-41, 1975.
5. Urrutia, J., Art Gallery and Illumination Problems, in Handbook on Computational Geometry, J. Sac, and J. Urrutia, (eds.), Elsevier Science Publishers, Amsterdam, 2000, p. 973-1027.
6. Fisk, S. "A Short Proof of Chvátal's Watchman Theorem." J. Combin. Th. Ser. B 24, 374, 1978.
7. Beigel, R.; Eppstein, D. (2005), "3-coloring in time O(1.3289n)", Journal of Algorithms, 54 (2)): 168–204
8. B. Chazelle and J. Incerpi, Triangulation and shape complexity, ACM Trans. on Graphics, vol. 3, pp.135-152, 1984.
9. B. Chazelle, Triangulating a simple polygon in linear time, Discrete Comput. Geom., vol. 6, pp. 485-524, 1991.
10. G.H. Meisters, Polygons have ears, Amer. Math. Monthly, vol. 82, pp. 648-651, 1975.
11. Barenboim, L.; Elkin, M. (2009), "Distributed (Δ + 1)-coloring in linear (in Δ) time", Proceedings of the 41st Symposium on Theory of Computing, pp. 111–120,