Évariste Math Club

Zero Prerequisite Contest

Introduction to Computational Geometry

Introduction and motivation

In a narrow sense *computational geometry* is concerned with computing geometric properties of sets of geometric objects in space such as the simple *above/below* relationship of a given point with respect to a given line. In a broader sense computational geometry is concerned with the *design* and *analysis* of algorithms for solving geometric problems. In a deeper sense it is the study of the inherent *computational complexity* of geometric problems under varying models of computation. In this latter sense it pre-supposes the determination of which geometric properties are computable in the first place.

Important applications of computational geometry include robotics (motion planning and visibility problems), geographic information systems (GIS) (geometrical location and search, route planning), integrated circuit design (IC geometry design and verification), computer-aided engineering (CAE) (mesh generation), computer vision (3D reconstruction).

Something interesting for the theory folks - Ketan Mulmuley and Milind Sohoni have presented an approach to the **P** versus **NP** problem through algebraic geometry, dubbed Geometric Complexity Theory, or GCT. They have reduced a question about the nonexistence of polynomial-time algorithms for all **NP**-complete problems to a question about the existence of a polynomial-time algorithm (with certain properties) for a specific problem.

In essence, they define a family of high-dimension polygons P_n based on group representations on certain algebraic varieties. Roughly speaking, for each n, if P_n contains an integral point, then any circuit family for the Hamiltonian path problem must have size at least $n^{\log n}$ on inputs of size n, which implies P ≠ NP. Thus, to show that P ≠ NP it suffices to show that P_n contains an integral point for all n.

So much for motivating the study of computational geometry. Let us look at an interesting problem:

The Art Gallery Problem

An art gallery has several rooms. And each room is guarded by cameras. Our goal is to place cameras i.e find a ‘guarding set’ of cameras so as to render the whole art gallery, every nook and cranny, visible to a camera.

Some basic definitions:

- Polygon Curve: A polygonal curve is a finite sequence of line segments, called *edges*, joined end-to-end. The endpoints of the edges are called *vertices*.

- Closed Polygon Curve: let $v_0, v_1, \ldots, v_n$ denote the set of n + 1 vertices, and let e_1, $e_2, \ldots, e_n$ denote a sequence of n edges, where $e_i = (v_{i-1}, v_i)$. A polygonal curve is ***closed*** if $v_0 = v_n$.
- Simple Polygon Curve: A polygonal curve is ***simple*** if it is not self-intersecting. More precisely this means that each edge e_i does not intersect any other edge, except for the endpoints it shares with its adjacent edges.
- Polygonal chains: a **polygonal chain** is a connected series of line segments.
- Convex Polygon: A **convex polygon** is defined as a polygon with all its interior angles less than 180°.

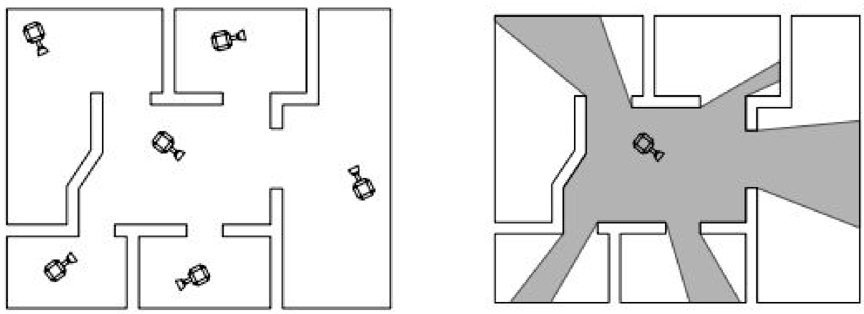


Figure 1: A 'floor plan' for the art gallery

Let's formalize the notion of gallery. A gallery is, of course, a 3-dimensional space, but a floor plan gives us enough information to place the cameras. Therefore we model a gallery as a polygonal region in the plane. We further restrict ourselves to regions that are simple polygons, that is, regions enclosed by a single closed polygonal chain that does not intersect itself. Thus we do not allow regions with holes. A camera position in the gallery corresponds to a point in the polygon. A camera *sees* those points in the polygon to which it can be connected with an open segment that lies in the interior of the polygon. How many cameras do we need to guard a simple polygon?
Reasonably, it would depends on the polygon at hand: the more complex the polygon, the more cameras are required. We shall therefore express the bound on the number of cameras needed in terms of *n*, the number of vertices of the polygon. But even when two polygons have the same number of vertices, one can be easier to guard than the other. A convex polygon, for example, can always be guarded with one camera. (Is it clear why that is true?)
To be on the safe side we shall look at the worst-case scenario, that is, we shall give a bound that is good for any simple polygon with n vertices. (It would be nice if we could find the minimum number of cameras for the specific polygon we are given, not just a worst-case bound. Unfortunately, the problem of finding the minimum number of cameras for a given polygon is '**NP-hard'** - if you don't know what that means, read about it after the ZPC.)

Let P be a simple polygon with n vertices. Because P may be a complicated shape, it seems difficult to say anything about the number of cameras we need to guard P. So maybe we should first decompose P into pieces that are easy to guard? The simplest such “easy to guard” shape is a triangle, and it seems like a reasonable conjecture that every simple polygon can be decomposed into triangles.

Some terms again:

- A diagonal is an open line segment that connects two vertices of P and lies in the interior of P.
- A decomposition of a polygon into triangles by a *maximal* set of non-intersecting diagonals is called a triangulation of the polygon - see Figure 2. (We require that the set of non-intersecting diagonals be maximal to ensure that no triangle has a polygon vertex in the interior of one of its edges. This could happen if the polygon has three consecutive collinear vertices.) Triangulations are usually not unique; the polygon in Figure 2, for example, can be triangulated in many different ways.

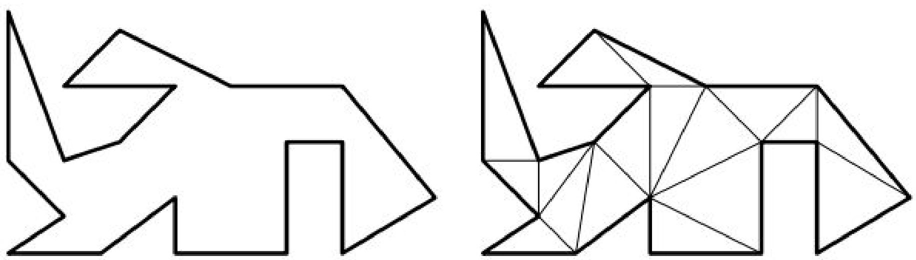
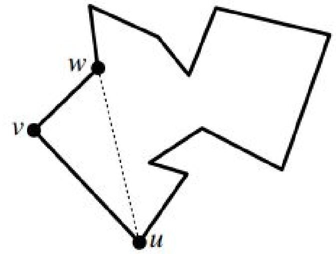


Figure 2

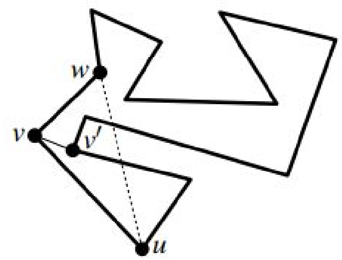
We can guard P by placing a camera in every triangle of a triangulation T_P of P. But does a triangulation always exist? And how many triangles can there be in a triangulation? The following theorem answers these questions-

Theorem 1: Every simple polygon admits a triangulation, and any triangulation of a simple polygon with n vertices consists of exactly n−2 triangles.
Proof.

We prove this theorem by induction on n. When n = 3 the polygon itself is a triangle and the theorem is trivially true. Let n > 3, and assume that the theorem is true for all m < n. Let P be a polygon with n vertices.
We first prove the existence of a diagonal in P. Let v be the leftmost vertex of P. (In case of ties, we take the lowest leftmost vertex.) Let u and w be the two neighboring vertices of v on the boundary of P. If the open segment $\overline{uw}$ lies in the interior of P, we have found a diagonal.



Otherwise, there are one or more vertices inside the triangle defined by u, v, and w, or on the diagonal $\overline{uw}$. Of those vertices, let v' be the one farthest from the line through u and w.

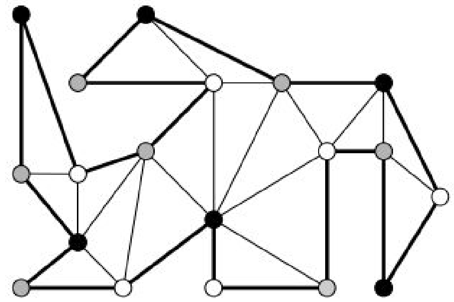


The segment connecting v' to v cannot intersect an edge of P, because such an edge would have an endpoint inside the triangle that is farther from the line through u and w, contradicting the definition of v'. Hence, $\overline{vv'}$ is a diagonal.
So a diagonal exists. Any diagonal cuts P into two simple sub-polygons P_1 and P_2. Let m_1 be the number of vertices of P_1 and m_2 the number of vertices of P_2. Both m_1 and m_2 must be smaller than n, so by induction P_1 and P_2 can be triangulated. Hence, P can be triangulated as well. It remains to prove that any triangulation of P consists of n−2 triangles. To this end, consider an arbitrary diagonal in some triangulation T_P. This diagonal cuts P into two sub-polygons with m_1 and m_2 vertices, respectively. Every vertex of P occurs in exactly one of the two sub-polygons, except for the vertices defining the diagonal, which occur in both sub-polygons. Hence, $m_1+m_2 = n+2$. By induction, any triangulation of P_i consists of $m_i - 2$ triangles, which implies that T_P consists of $(m_1 - 2)+(m_2 - 2) = n-2$ triangles.

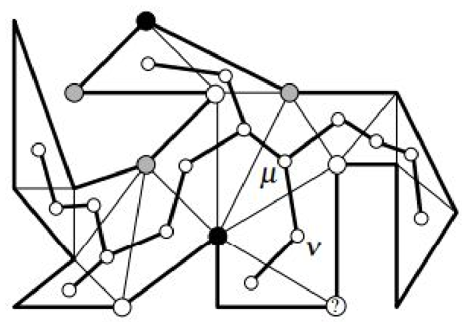
QED

Theorem 1 implies that any simple polygon with n vertices can be guarded with n−2 cameras. But placing a camera inside every triangle seems overkill. A camera placed on a diagonal, for example, will guard two triangles (since a diagonal is common to those two triangles), so by placing the cameras on well-chosen diagonals we might be able to reduce the number of cameras to roughly n/2.
But a vertex can be common to more than one triangles, right? Placing cameras at vertices seems even better, because a vertex can be incident to many triangles, and a camera at that vertex guards all of them. This suggests the following approach.

Let T_P be a triangulation of P. Select a subset of the vertices of P, such that any triangle in T_P has at least one selected vertex, and place the cameras at the selected vertices. To find such a subset we assign each vertex of P a color: white, gray, or black. The coloring will be such that any two vertices connected by an edge or a diagonal have different colors. This is called a *3-coloring* of a triangulated polygon. In a 3-coloring of a triangulated polygon, every triangle has a white, a gray, and a black vertex.



Hence, if we place cameras at all gray vertices, say, we have guarded the whole polygon. By choosing the smallest color class to place the cameras, we can guard P using at most [n/3] cameras, where [•] implies the greatest integer function.
But does a 3-coloring always exist? The answer is yes. To see this, we look at what is called the *dual graph*[1] of T_P. This graph $G(T_P)$ has a node for every triangle in T_P. We denote the triangle corresponding to a node ν by t(ν). There is an arc between two nodes ν and μ if t(ν) and t(μ) share a diagonal.



The arcs in $G(T_P)$ correspond to diagonals in T_P. Because any diagonal cuts P into two, the removal of an edge from $G(T_P)$ splits the graph into two. Hence, $G(T_P)$ is a tree. (Notice that this is not true for a polygon with holes.) This means that we can find a 3-coloring using a simple graph traversal. How? That is for you to answer in this ZPC.

We conclude that a triangulated simple polygon can always be 3-colored. As a result, any simple polygon can be guarded with [n/3] cameras. But perhaps we can do even better. After all, a camera placed at a vertex may guard more than just the incident triangles. Unfortunately, for any n there are simple polygons that require [n/3] cameras.
We just proved the Art Gallery Theorem, a classical result from combinatorial geometry.

Theorem 2 (Art Gallery Theorem): For a simple polygon with n vertices, [n/3] cameras are occasionally necessary and always sufficient to have every point in the polygon visible from at least one of the cameras.

[1] Dual of a graph: the dual graph of a plane graph G is a graph that has a vertex for each face of G.

ZPC
Problem Set

Total points : 20

Note: There might be questions that are not allotted points proportionate to their difficulty.

Q1. Give an example of a simple polygon that does not have a guarding set of less than [n/3] cameras.
2 points
Q2. A *rectilinear polygon* is a simple polygon of which all edges are horizontal or vertical. Let P be a rectilinear polygon with n vertices. Give an example to show that [n/4] cameras are sometimes necessary to guard it. ([x] represents the greatest integer function of x.)
2 points
Q3. Give the pseudo-code of the algorithm to compute a 3-coloring of a triangulated simple polygon.
3 points
Q4. Given a simple polygon P with n vertices and a point p inside it, describe a method to compute the region inside P that is visible from p.
3 points
Q5. Suppose that a simple polygon P with n vertices is given, together with a set of diagonals that partitions P into convex quadrilaterals. How many cameras are sufficient to guard P?
5 points
Q6. Can you come up with a non-convex polyhedra in 3D space such that even if you place a guard at every vertex there would still be points in the polygon that are not visible to any guard? Do explain your reason.
5 points