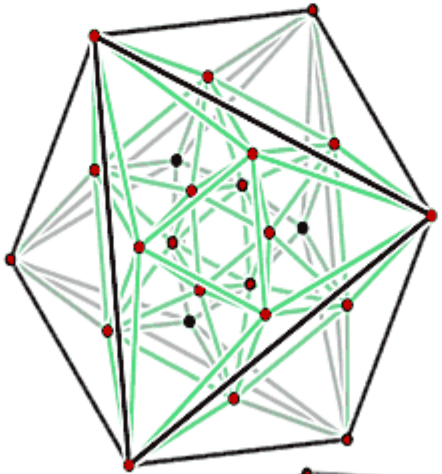
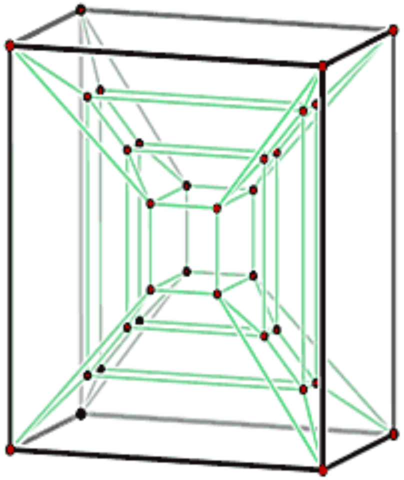
Flavor of Computational Geometry





Polygon Triangulation

Shireen Elhabian and Aly A. Farag

University of Louisville

February 2010

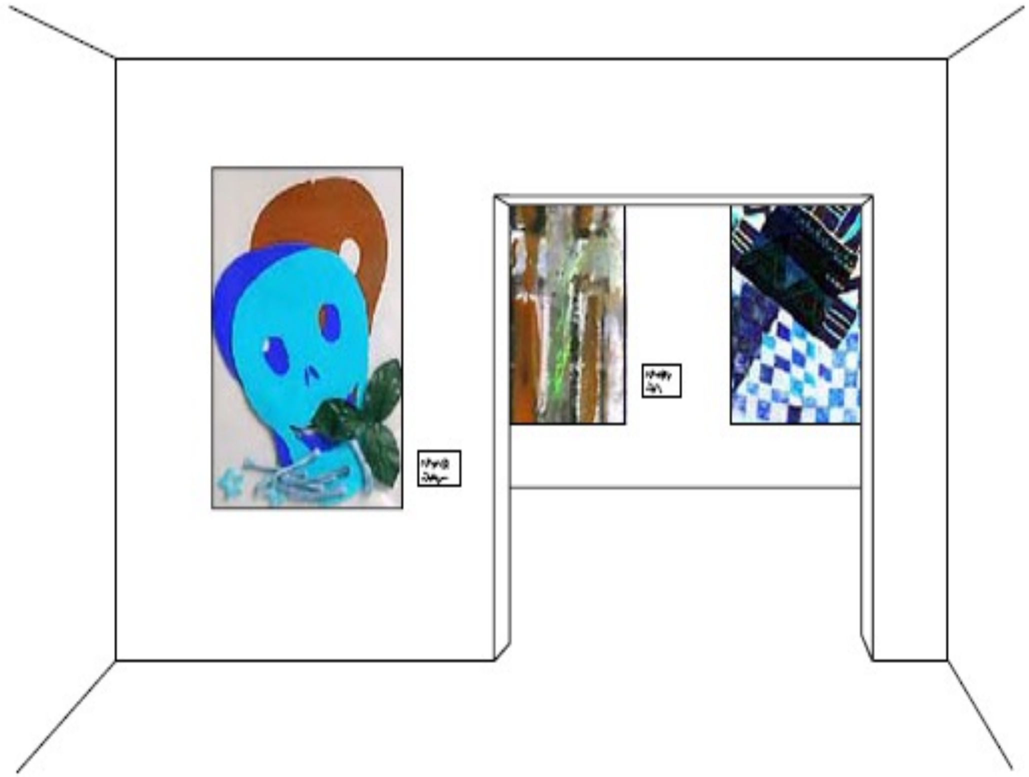
Agenda

- Overview
- Polygon Triangulation
  - Art Gallery Theorems
  - Area of Polygon
  - Segment Intersection
  - Finally … Triangulation

Overview – Computational Geometry

- Computational geometry is the study of algorithms for solving geometric problems on a computer.
- In this lecture series we are mainly concerned with discrete and combinatorial geometry where polygons play a much larger role than do curved boundaries.
- Much of the work of continuous curves and surface falls under the umbrella of geometric/solid modeling.
- Computational geometry is a field by itself; however we will be focusing on three main topics.
- We will begin with the core concern of computational geometry which is **polygon partitioning**, then move to the issue of **convex hull** computation and finalizing with **triangulation of a given set of points**.
- In this lecture, we will discuss **polygon partitioning/triangulation**.

Art Gallery Theorems

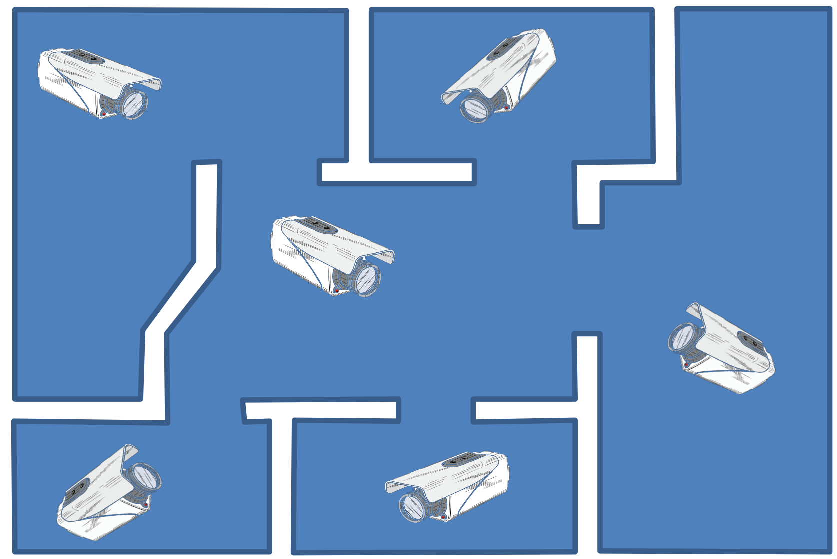


Art Gallery Problem

- Imagine an art gallery room,the question now is:
  - **How many stationary guards are needed to guard the room?**
- Each guard is considered to be a fixed point that can see in every direction, that is, has a 2π range of visibility.
- An equivalent formulation is to ask how many point lights are needed to fully illuminate the room.
- This problem can also reformulated as how many video cameras required to guard this gallery?
- These cameras are usually hung from the ceiling and they rotate about a vertical axis. The images taken from these cameras are sent to TV screens in the office of the night watch.
- Since it is easier to keep an eye on few TV screens, the number of cameras should be as small as possible. This small number will lessen the cost of the security system needed to watch-out this place. However on the other hand we cannot have too few cameras, because every part of the gallery should be visible to at least one of the cameras.
- Hence we should place the cameras at strategic positions such that each of the cameras guards a large part of the gallery. This gives rise to what-so-called **Art Gallery Problem**.

Definition 1: Art Gallery Problem: *how many cameras/guards do we need to guard a given gallery and how do we decide where to place them?*

- Let's now define the art gallery problem more precisely, the notion of gallery can be formalized as follows;
  - A gallery is a 3-dimensional space, however its floor plan gives us enough information to place the cameras, hence we can model a gallery as a polygonal region in the 2-dimensional space.



- Most computational geometry algorithms perform their work on geometric objects known as polygons. Polygons are considered to be a convenient way to represent many real-world objects, however polygons can rather be complicated such that they are needed to be composed of simpler components, this leads to the topic of this section: partitioning polygons.

- In the following we will formalize the definition of a polygon, specifying which types of polygon we will deal with. Intuitively, we will restrict ourselves with polygonal regions which have no holes.

Polygon Definition

Definition 2: *A* **polygon** *is the region of a plane bounded by a finite collection of line segments forming a simple closed curve which is homeomorphic image of a disk, i.e. it is a certain deformation of a disk.*

A more formal definition can be given as follows,

Definition 3: *Let* $v_0, v_1, \dots, v_{n-1}$ *be* n *points in the plane* $\mathbb{R}^2$, *here all index arithmetic will be* $\bmod n$, *implying a cyclic ordering of the points, with* v_0 *following* v_{n-1} *since* $(n-1)+1 \equiv n \equiv 0 (\bmod n)$. *Let* $e_0 = v_0v_1$, $e_1 = v_1v_2$, ... , $e_{n-1} = v_{n-1}v_0$ *be* n *segments connecting the points. Then these segments bound a polygon (simple closed curve) if and only if:*

(1) The intersection of each pair of adjacent segments in the cyclic ordering is the single point shared between them, i.e. $e_i \cap e_{i+1} = v_{i+1}$, *for all* $i = 0,1,2,\dots,n-1$.

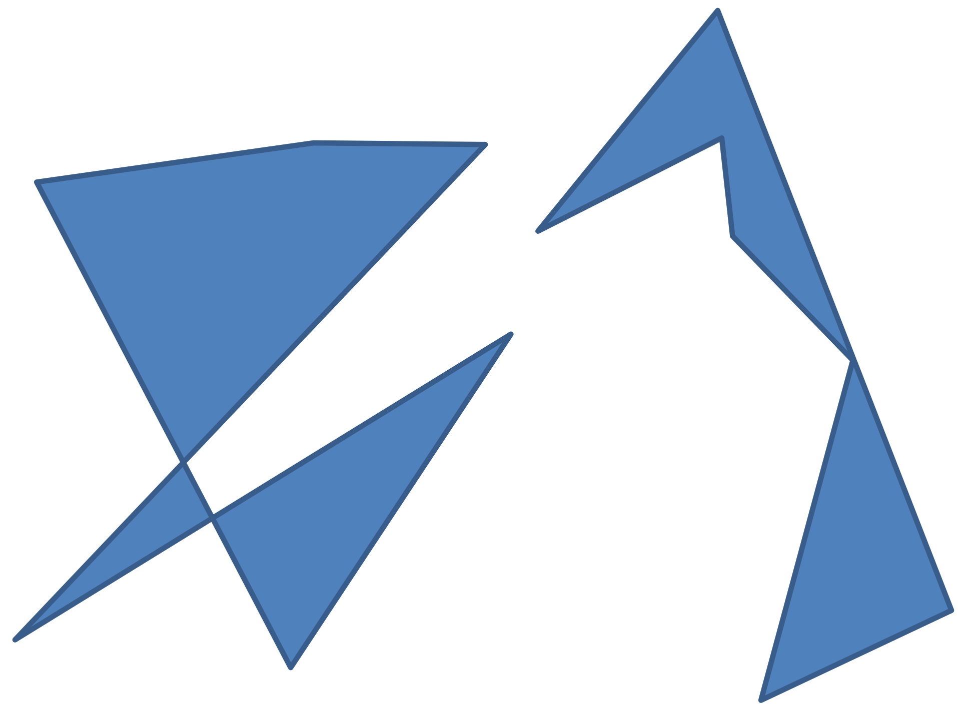
(2) Nonadjacent segments do not intersect, i.e. $e_i \cap e_j = \phi$, *for* $j \neq i+1$.

The reason these segments define a curve is that they are connected end to end, the reason the curve is closed is that they form a cycle, the reason the closed curve is simple is that nonadjacent segments do not intersect. The points v_i *are called* **vertices** *of the polygon and the segments* e_i *are called its* **edges**. *Note that a polygon with n vertices has n edges.*

Theorem 1: Jordan Curve Theorem : *Every simple close plane curve divides the plane into two components.*

The two parts of the plane are called the interior and exterior of the curve. The exterior is unbounded while the interior is bounded. Note that we define a polygon $\mathcal{P}$ as a closed region of the plane, however it is common to consider the polygon to be just the segments bounding the region and not the region itself. we will use the notion $\partial\mathcal{P}$ to denote the boundary of the polygon $\mathcal{P}$, hence by our definition we have $\partial\mathcal{P} \subseteq \mathcal{P}$.

We will use the convention of listing the vertices of a polygon in a counterclockwise order such that if you walked along the boundary of the polygon visiting the vertices in that order, the interior of the polygon would be always to your left.



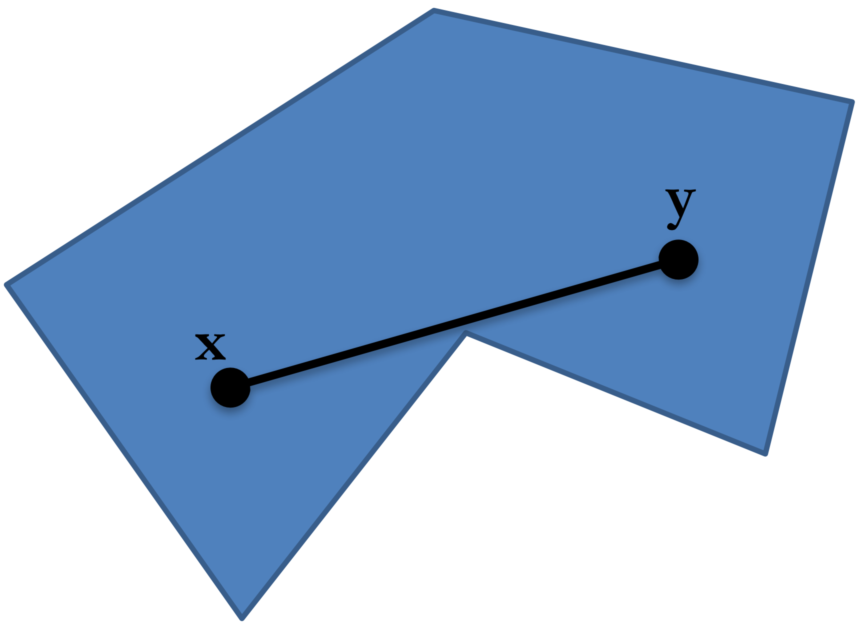
Examples of non-simple polygons, where adjacent segments share a common point, however non-adjacent segments intersect

Visibility

It is time now to define the notion of visibility.

Definition 4: *Let x, y be two points in the polygon $\mathcal{P}$, we say that point x can see point y, or y is* **visible** *to x if and only if the closed segment xy is interior to the polygon $\mathcal{P}$, i.e. $xy \subseteq \mathcal{P}$. Equivalently, we say x has* **clear visibility** *to y if $xy \subseteq \mathcal{P}$ and $xy \cap \partial\mathcal{P} \subseteq \{x, y\}$, that is the segment joining the points x, y intersect the boundary of the polygon $\mathcal{P}$ at either one of them or both.*

Definition 5: *A guard/camera is a point. A set of guards/cameras is said to* **cover** *a polygon if every point in the polygon is visible to some guard/camera.*



Point *x* can see point *y* since the line segment joining these two points is totally enclosed in the interior region of the polygon.

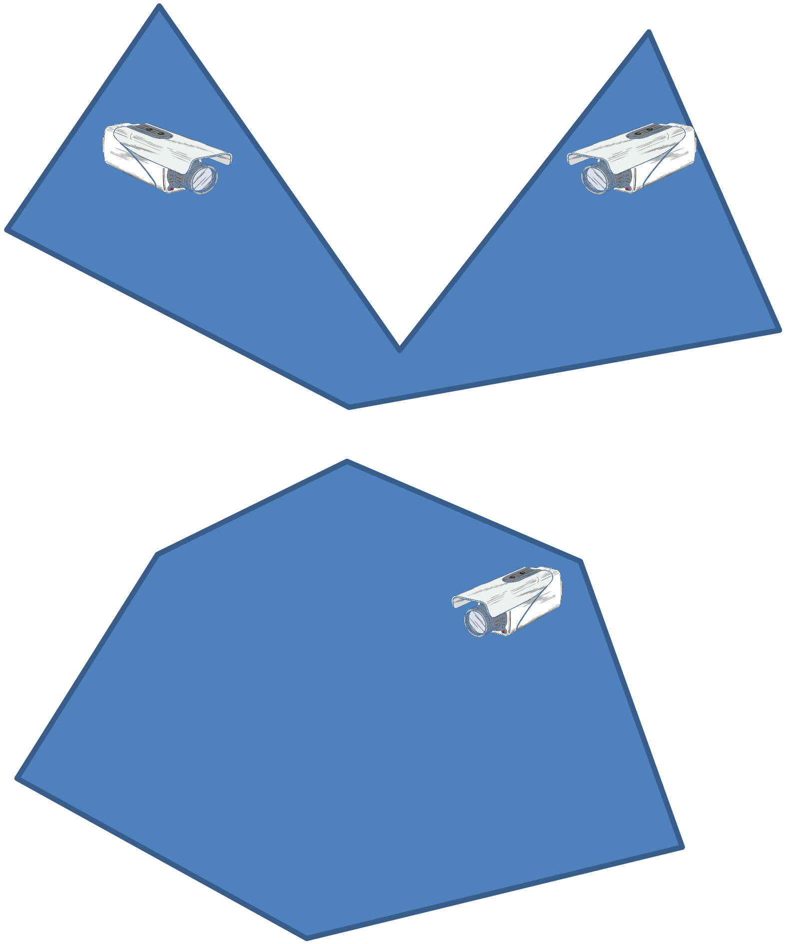
How many Cameras/Guards?!!

- How many cameras/guards do we need to guard a simple polygon?

- This mainly depends on the polygon at hand, since the more complex the polygon is , the more cameras/guards required.

- However we can express the lower and upper bounds on the number of cameras in terms of the number of vertices of the polygon, i.e. . It is important to note that even if we have two polygons with the same number of vertices, one might be easier to guard than the other.

- A *convex polygon* for instance can be guarded with one camera.



Two polygons with the same number of vertices, however the one on the top needs at least two cameras to be guarded since it is non-convex polygon while the one on the bottom needs only one camera because it is a convex polygon

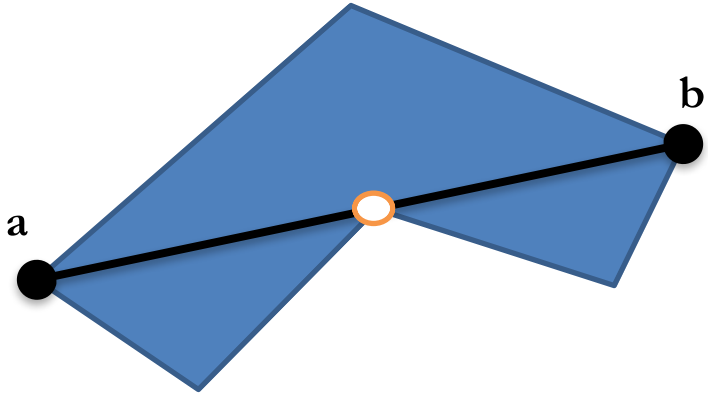
Definition 6: *A polygon is said to be* **convex** *if it has no dents. A vertex is called* **reflex vertex** *if its internal angle is strictly greater than* π, *otherwise the vertex is called* **convex vertex**. *A polygon with* n *vertices is said to be* **convex polygon** *if all its vertices are convex.*

Looking at the worst-case scenario, what is the bound of the number of cameras/guards that is good for any simple polygon with n vertices.

However since the polygon can be arbitrary complex, we need to decompose it into pieces that are easy to guard, namely triangles. This gives rise to the theory of *polygon triangulation.*

Decomposing a polygon into triangles can be done by drawing *diagonals* between pairs of polygon vertices.

Definition 7: *A* **diagonal** *of a polygon* $\mathcal{P}$ *is a line segment between two of its vertices* a *and* b *that are clearly visible to one another, that is the intersection of the closed segment* ab *with the boundary of the polygon* $\partial\mathcal{P}$ *is exactly the set* $\{a, b\}$, *thus the diagonal cannot make grazing contact with the boundary, in other words a diagonal is an open line segment that connects two vertices of* $\mathcal{P}$ *and at the same time the whole diagonal lies in the interior of the polygon* $\mathcal{P}$.



The closed segment *ab* is not a diagonal since it makes a grazing contact with the boundary

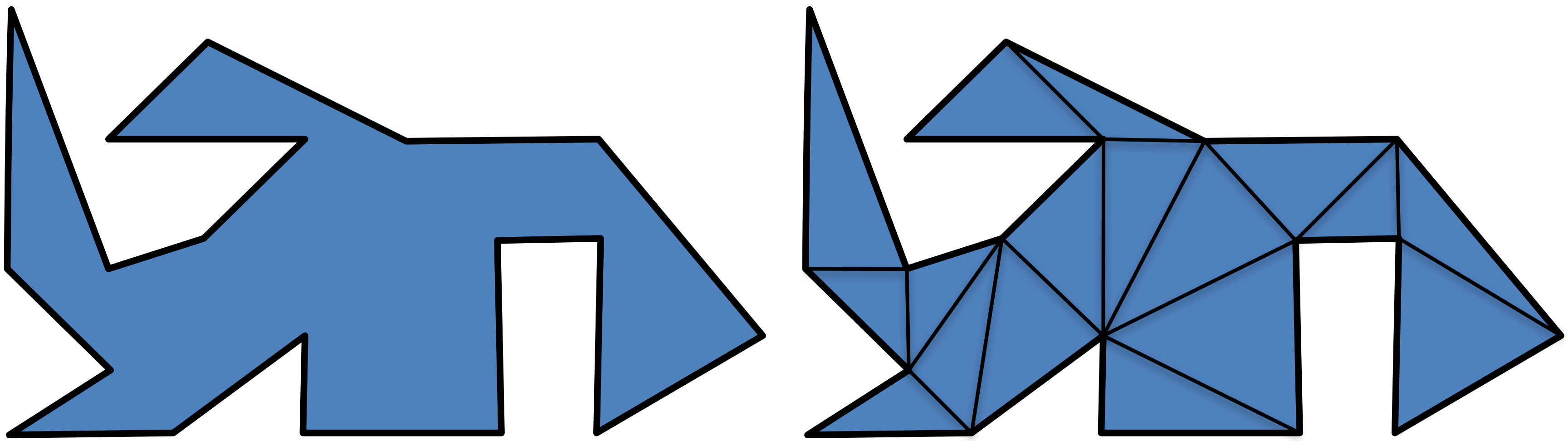
In order to partition a given polygon into triangles, we need to have non-crossing/non-intersecting diagonals, which can be defined as follows.

Definition 8: *Two diagonals are said to be* **non-crossing/non-intersecting** *diagonals if their intersection is a subset of their endpoints, that is they share no interior points.*

If we add as many non-crossing diagonals to a polygon as possible, the interior of the polygon is partitioned into triangles. Such a partition is called a *triangulation* of a polygon.

Definition 9: *A decomposition of a polygon $\mathcal{P}$ into triangles by a maximal set of non-intersecting/non-crossing diagonal is called* **triangulation** $T_{\mathcal{P}}$ *of the polygon $\mathcal{P}$. 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, i.e. a given polygon can have more than one triangulations, this is because diagonals may be added in an arbitrary order, as long as they are legal diagonals and non-crossing.



Simple polygon and one of its possible triangulations

Now, we can guard $\mathcal{P}$ by placing a camera in every triangle of a triangulation $T_{\mathcal{P}}$ of $\mathcal{P}$. However, does a triangulation always exist? And how many triangles can there be in a triangulation? The following theorem answers these questions.

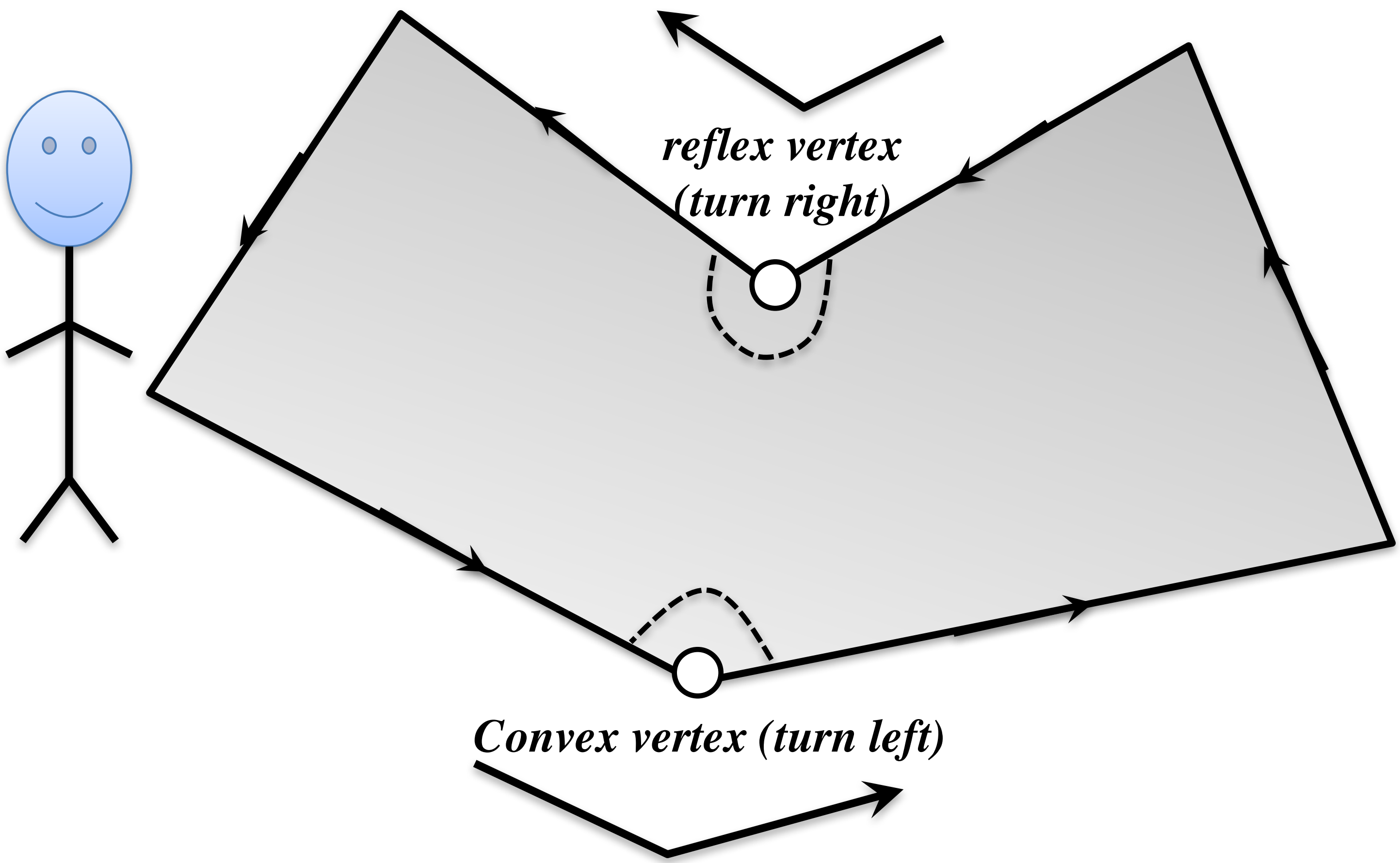
Theorem 2 - Triangulation: *Every simple polygon admits a triangulation, and any triangulation of a simple polygon with n- vertices consists exactly n-2 triangles.*

The key to proving the existence of a triangulation is proving the existence of a diagonal. Once we have that, the rest will follow easily. However, we need a fact to begin the proof: Every polygon must have at least one strictly convex vertex.

Lemma 1: *Every polygon must have at least one strictly convex vertex.*

Proof:

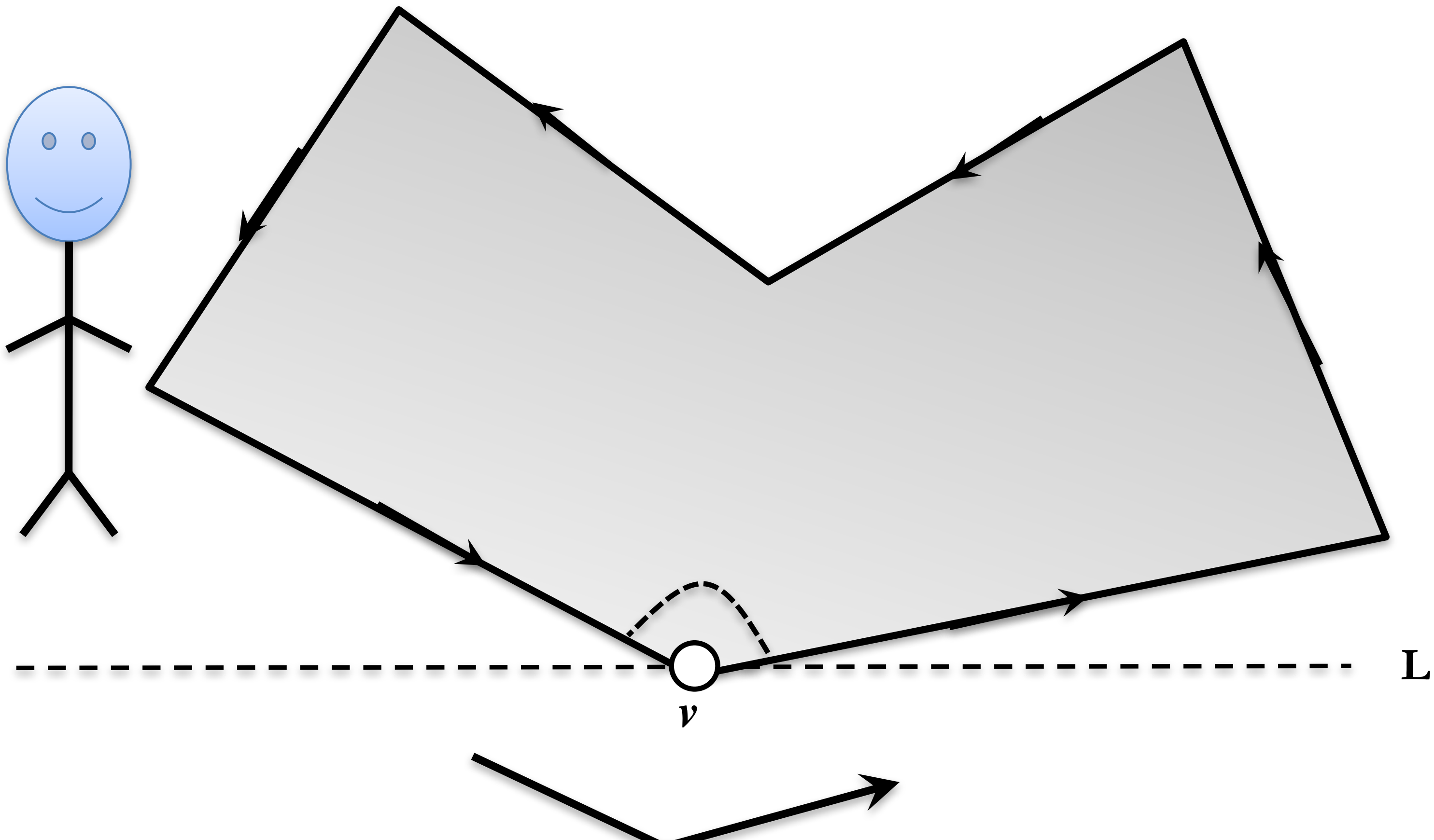
Let the edges of a polygon $\mathcal{P}$ are oriented so that their direction indicates a counterclockwise traversal, then a strictly convex vertex is a left turn for someone walking around the boundary, and a reflex vertex is a right turn. The interior of the polygon is always to the left of this hypothetical walker.



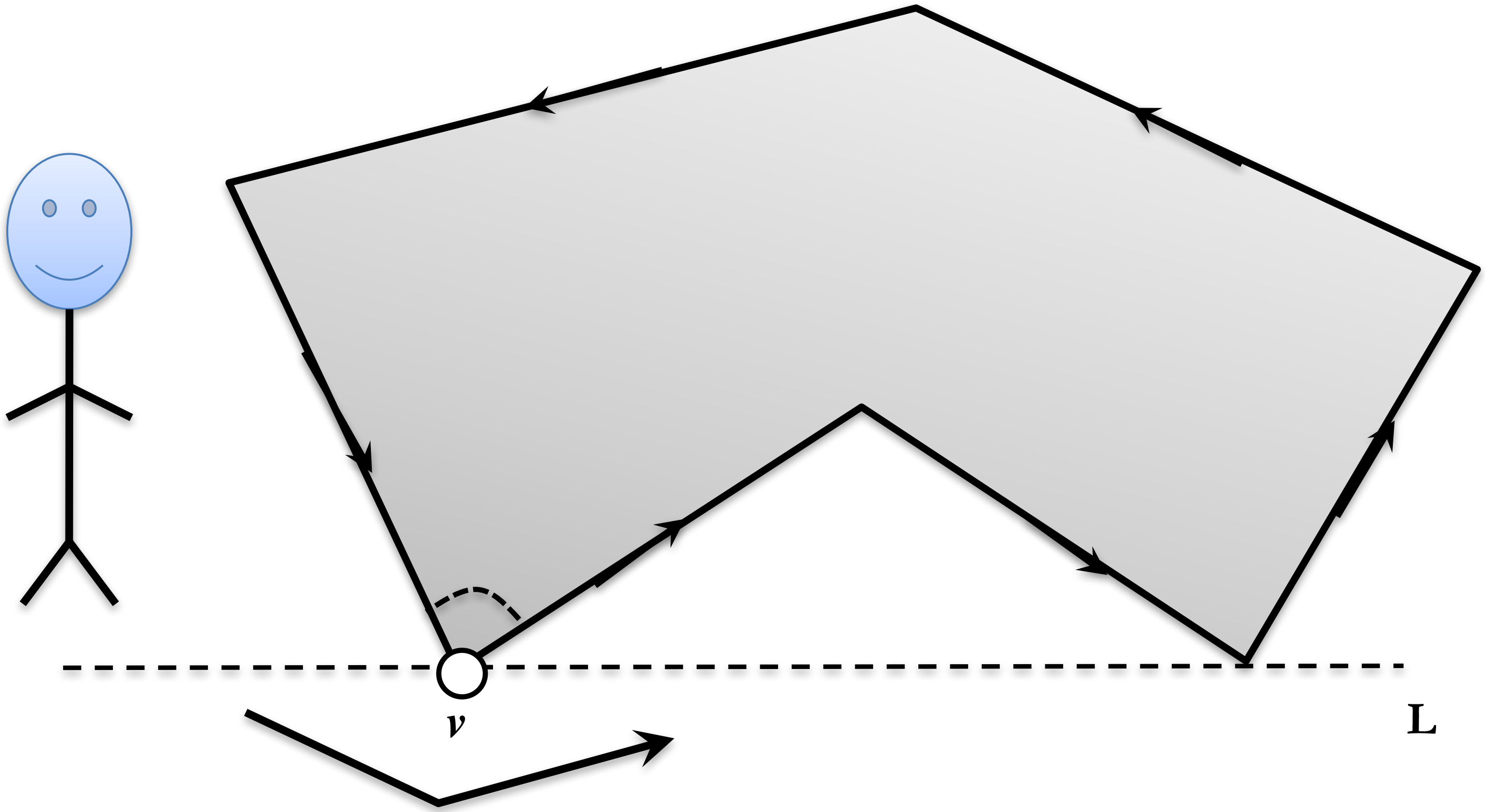
Proof: Continued

Let L be a line through a lowest vertex v of $\mathcal{P}$, lowest in having minimum y coordinate with respect to a coordinate system; if there are several lowest vertices, let v be the rightmost. The interior of $\mathcal{P}$ must be above L. The edge following v must lie above L. Together these conditions imply that the walker makes a left turn at v and therefore that v is a strictly convex vertex.

□



The rightmost lowest vertex must be strictly convex.



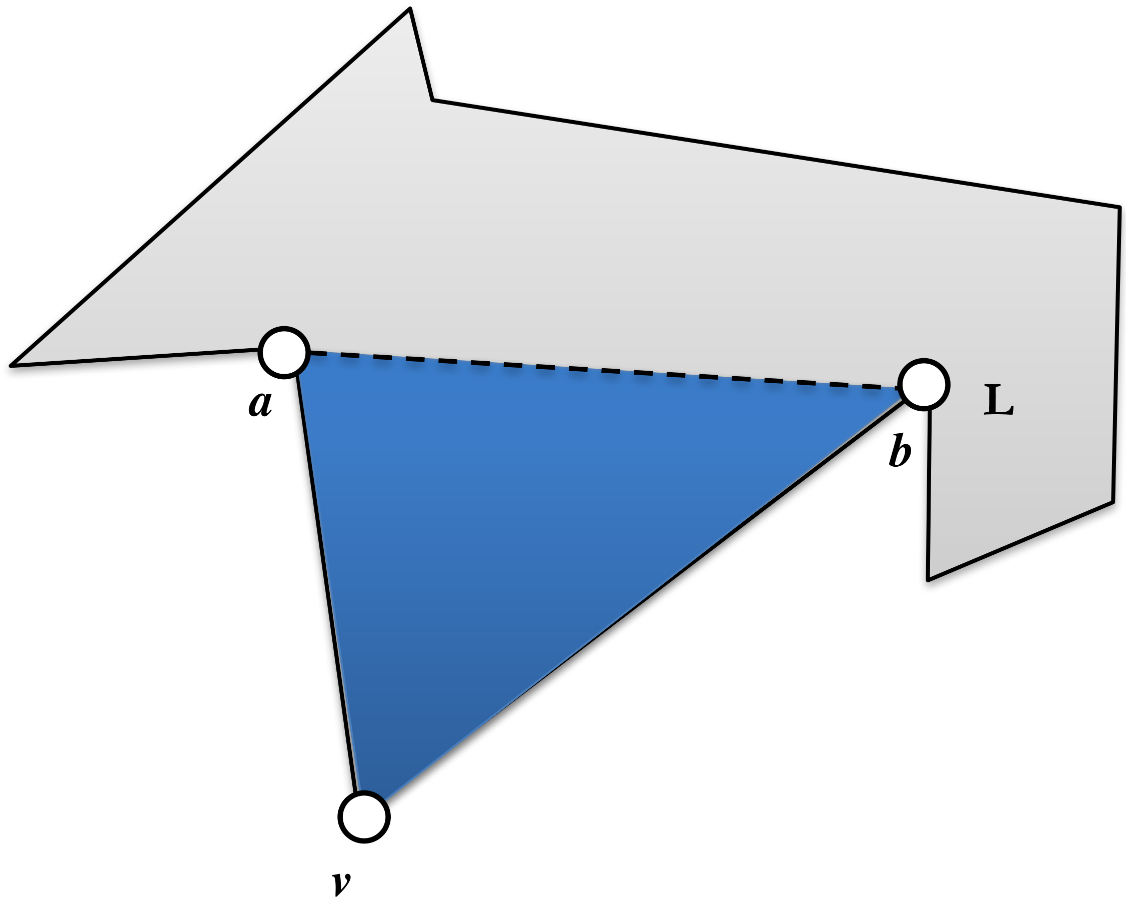
The proof still holds even if we turned the polygon upside down.

This proof can be used to construct an efficient test for the orientation of a polygon.

Lemma 2 - Meisters: *Every polygon of* $n \geq 4$ *vertices has a diagonal.*

Proof:

From Lemma 1, any given polygon must have at least one strictly convex vertex, hence let *v* be a strictly convex vertex. Let *a* and *b* the vertices adjacent (neighbor) to *v*. If *ab* is a diagonal, we are finished.



Let *v* be a strictly convex vertex. Let *a* and *b* the vertices adjacent (neighbor) to *v*. Here the segment joining a and b constructs a diagonal, hence the Lemma holds.

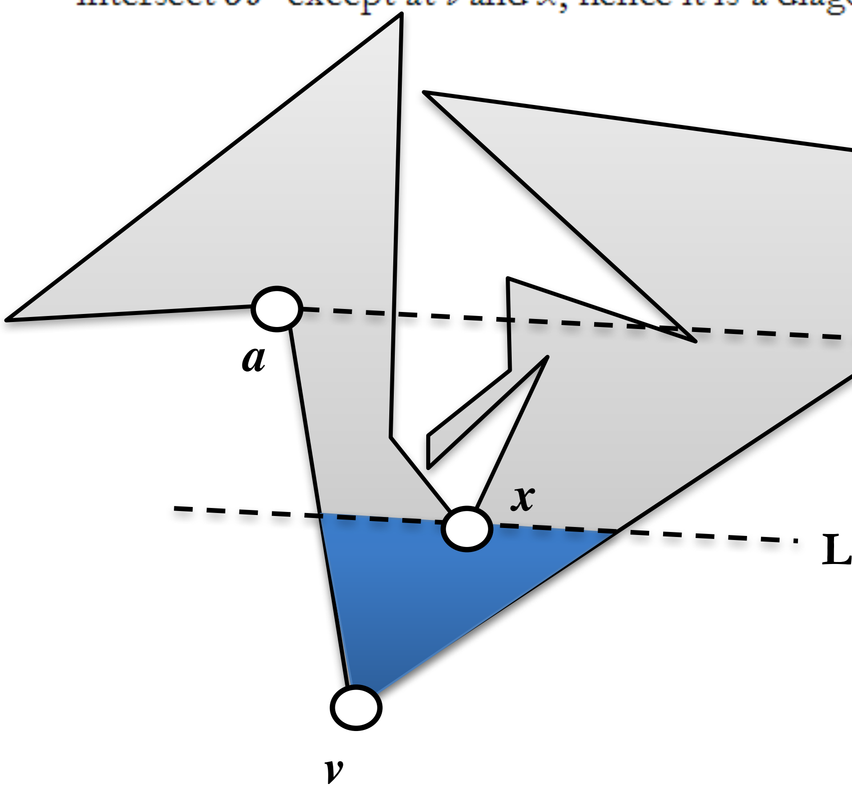
Proof: Continued

Suppose ab is not a diagonal. Then either ab is exterior to the polygon $\mathcal{P}$, or it intersects the boundary $\partial\mathcal{P}$. In either case, since $n > 3$, the closed triangle Δavb contains at least one vertex of $\mathcal{P}$ other than a, v, b (which are three of the polygon vertices).

Let x be the vertex of $\mathcal{P}$ in Δavb that is closest to v, where distance is measured orthogonal to the line through ab. Thus x is the first vertex in Δavb hit by a line L parallel to ab moving from v to ab.

It is clear that the interior of Δavb intersected with the half-plane bounded by L that includes v (the shaded region in the following figure) is empty of points on the boundary $\partial\mathcal{P}$. Therefore vx cannot intersect $\partial\mathcal{P}$ except at v and x, hence it is a diagonal.

$\square$



The segment joining a and b intersects the boundary of the polygon, hence the triangle avb contains at least one vertex of the polygon, let x be the closest vertex to v, hence *vx* must be a diagonal

Although there can be different ways to triangulate the same polygon, however all triangulations of the same polygon share same number of diagonals and same number of triangles.

Lemma 3 - Number of diagonals: *Every triangulation of a polygon $\mathcal{P}$ of n vertices uses $n-3$ diagonals and consists of $n-2$ triangles.*

Proof:

The proof goes by induction with the base step $n = 3$ which is the case of a triangle, it has $n - 2 = 1$ triangles with $n - 3 = 0$ diagonals interior to it. Now let $n > 3$, consider an arbitrary diagonal ab in some triangulation $T_{\mathcal{P}}$. This diagonal cuts $\mathcal{P}$ into two sub-polygons with m_1 and m_2 vertices, respectively. Every vertex of $\mathcal{P}$ occurs in exactly one of the two sub-polygons, except for the vertices defining the diagonal, which occur in both sub-polygons. Hence, $m_1 + m_2 = n + 2$, this is because a and b are counted in both m_1 and m_2. By induction, any triangulation of $\mathcal{P}_i$ consists of $m_i - 2$ triangles, which implies that $T_{\mathcal{P}}$ consists of $(m_1 - 2) + (m_2 - 2) = n - 2$ triangles, at the same time there are $(m_1 - 3) + (m_2 - 3) + 1 = n - 3$ diagonals with the final $+1$ term is counting for the diagonal ab.

Corollary 1 - Sum of Angles: *The sum of the internal angles of a polygon $\mathcal{P}$ of n vertices is $(n-2)\pi$.*

Proof:

There are $n - 2$ triangles by Lemma 3, and each contributes π to the internal angles.

□

Theorem 2 - Triangulation: *Every simple polygon admits a triangulation, and any triangulation of a simple polygon with n- vertices consists exactly n-2 triangles.*

Now, we are ready to prove it …

Proof:

We prove this theorem by induction on n. When $n = 3$, the polygon is a triangle and the theorem is true, that is the polygon is already triangulated and its triangulation contains $1 = n - 2$ triangles.

Now, let $n > 3$, assume that the theorem is true for all $m < n$, that is a polygon of m vertices can be triangulated with m-2 triangles, where $m < n$. Let $\mathcal{P}$ be a polygon with n-vertices, we need to prove that this polygon can be triangulated (given that all polygon with $m<n$ vertices can be triangulated).

Let $d = ab$ be a diagonal of $\mathcal{P}$, using Lemma 2 the existence of d is guaranteed. Since d be definition only intersects the boundary of $\mathcal{P}$ at its end points, this diagonal will cut $\mathcal{P}$ into two simple sub-polygons $\mathcal{P}_1$ and $\mathcal{P}_2$. Let m_1 be the number of vertices of $\mathcal{P}_1$ and m_2 the number of vertices of $\mathcal{P}_2$ Both $m_1$and m_2 must be smaller than n, so by induction $\mathcal{P}_1$ and $\mathcal{P}_2$ can be triangulated. Hence, $\mathcal{P}$ can be triangulated as well.

$\square$

Cameras/Guards Locations !!!

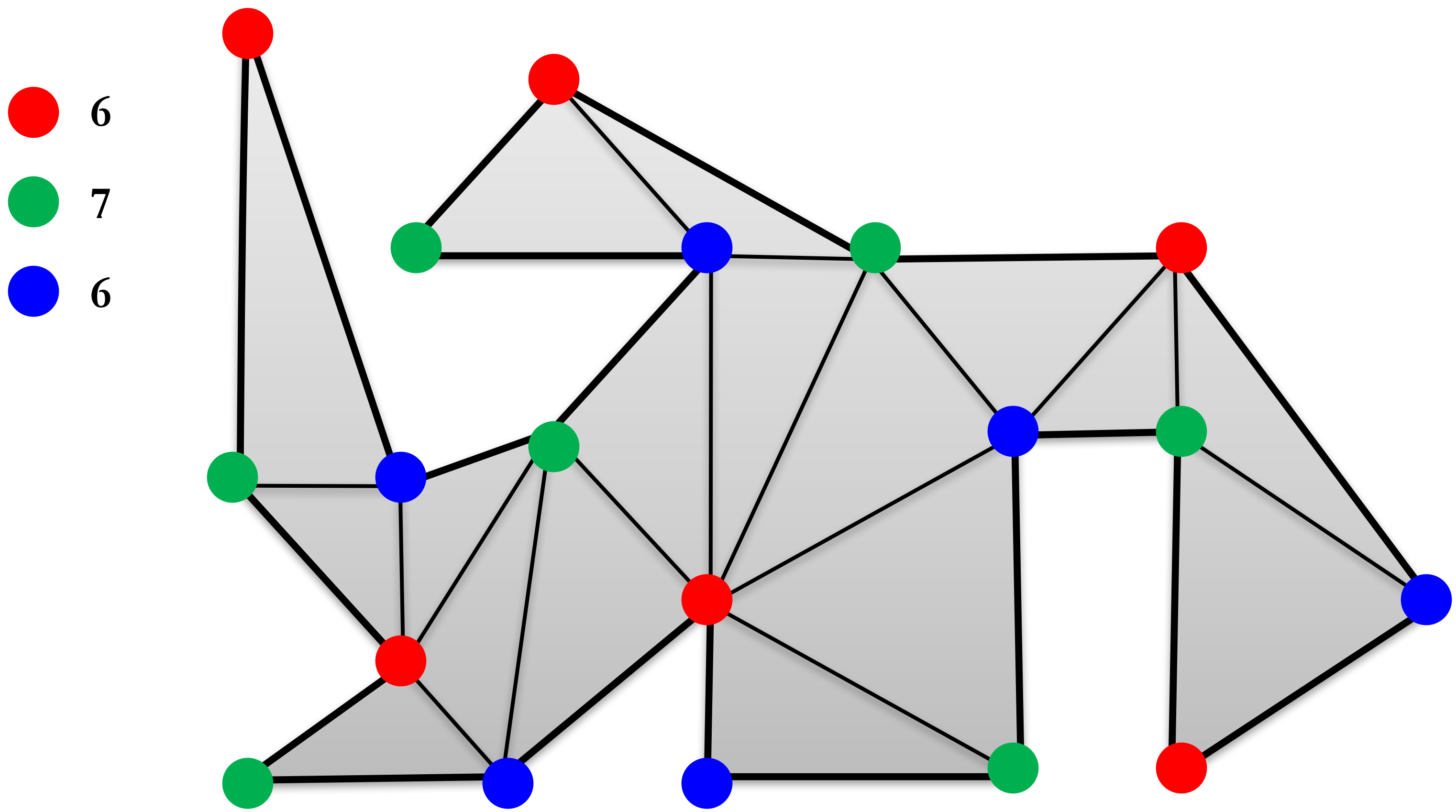
- Theorem 2 implies that any simple polygon with *n* vertices can be guarded with *n-2* cameras, however it didn't answer the question where to put these cameras.

- If we place a camera on a diagonal, such camera would be able to guard two triangles, hence by placing the cameras on well-chosen diagonal we might be able to reduce the number of cameras to roughly *n/2*.

- However placing cameras at polygon vertices seems to be much better because a vertex can belong to more than one triangle, and since a camera has 2π visibility, it can guard all these triangles.

- This gives rise to the following scheme:

Let $T_{\mathcal{P}}$ be a triangulation of $\mathcal{P}$. Select a subset of the vertices of $\mathcal{P}$, such that any triangle in $T_{\mathcal{P}}$ has at least one selected vertex, and place the cameras at the selected vertices. To find such a subset we assign each vertex of $\mathcal{P}$ a color: red, green and blue. 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 where every triangle has a red, a green and a blue vertex. Hence, if we place cameras at all red vertices, say, we have guarded the whole polygon. By choosing the smallest color class to place the cameras, we can guard $\mathcal{P}$ using at most $\left\lfloor \frac{n}{3} \right\rfloor$ cameras.



3-coloring of a simple polygon with 19 vertices it can be guarded by 6 cameras either place at the red or blue vertices

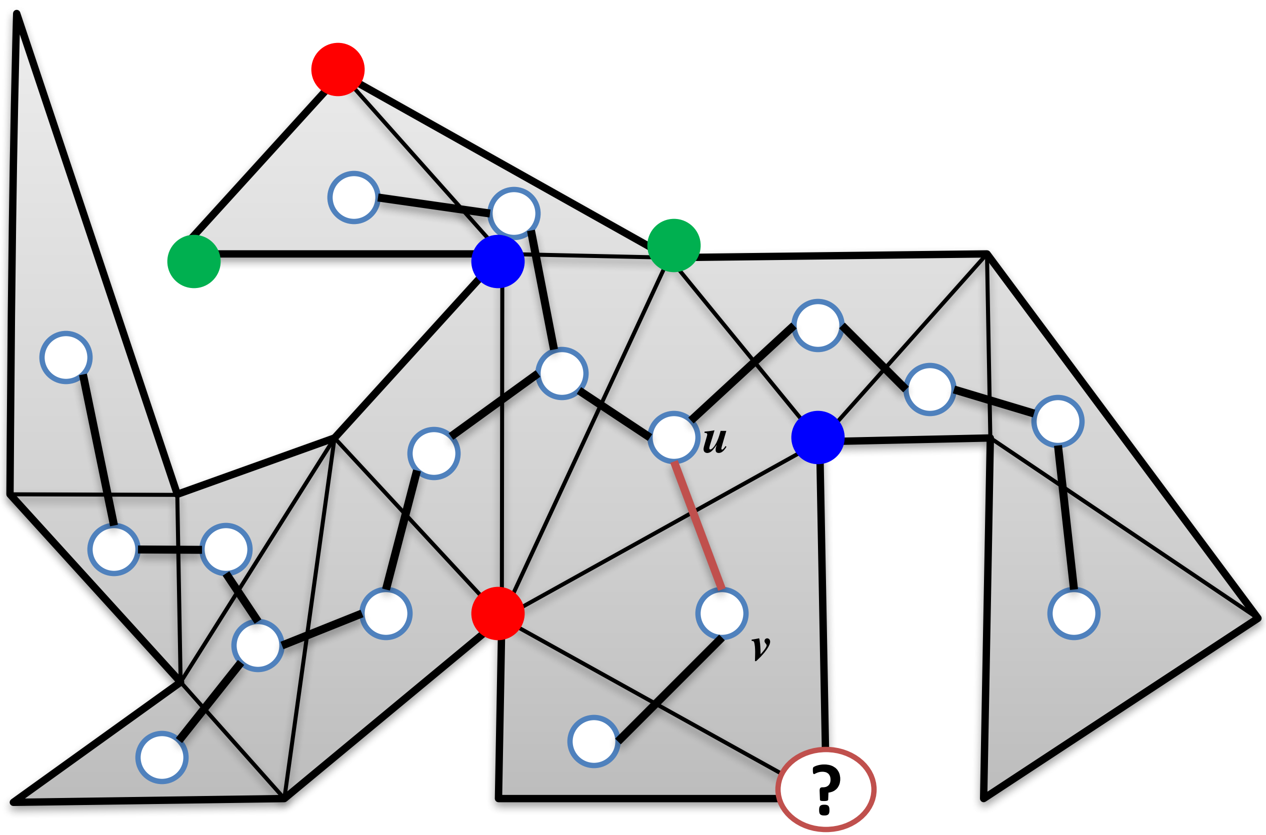
However, do we guarantee the existence of a 3-coloring of an arbitrary simple polygon? To answer this question we need to investigate what is called the *dual graph* of $T_{\mathcal{P}}$.

This graph $\mathcal{G}(T_{\mathcal{P}})$ has a node for every triangle in $T_{\mathcal{P}}$. Let's denote the triangle corresponding to a node v by $t(v)$. There is an arc between two nodes v and u if $t(v)$ and $t(u)$ share a diagonal, hence the arcs in $\mathcal{G}(T_{\mathcal{P}})$ correspond to diagonals in $T_{\mathcal{P}}$.

Because any diagonal cuts $\mathcal{P}$ into two sub-diagonals, the removal of an arc/edge from $\mathcal{G}(T_{\mathcal{P}})$ splits the graph into two sub-graphs. Hence, $\mathcal{G}(T_{\mathcal{P}})$ 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, such as depth first search.

The depth first search can be started from any node of $\mathcal{G}(T_{\mathcal{P}})$; the three vertices of the corresponding triangle are colored red, green and blue. Now suppose that we reach a node v in $\mathcal{G}(T_{\mathcal{P}})$, coming from node u. Hence, $t(v)$ and $t(u)$ share a diagonal. Since the vertices of $t(u)$ have already been colored, only one vertex of $t(v)$ remains to be colored. There is one color left for this vertex, namely the color that is not used for the vertices of the diagonal between $t(v)$ and $t(u)$. Because $\mathcal{G}(T_{\mathcal{P}})$; is a tree, the other nodes adjacent to v have not been visited yet, and we still have the freedom to give the vertex the remaining color.

Definition 10: *A* **k-coloring** *of a graph* $\mathcal{G}(T_{\mathcal{P}})$ *is an assignment of k colors to the nodes of the graph, such that no two nodes connected by an arc are assigned the same color.*



The *dual graph* $\mathcal{G}(T_{\mathcal{P}})$ of $T_{\mathcal{P}}$ has a node for every triangle in $T_{\mathcal{P}}$. Let's denote the triangle corresponding to a node v by $t(v)$. There is an arc between two nodes v and u if $t(v)$ and $t(u)$ share a diagonal, hence the arcs in $\mathcal{G}(T_{\mathcal{P}})$ correspond to diagonals in $T_{\mathcal{P}}$. Now suppose that we reach a node v in $\mathcal{G}(T_{\mathcal{P}})$, coming from node u. Since the vertices of $t(u)$ have already been colored, only one vertex of $t(v)$ remains to be colored. There is one color left for this vertex, namely the color that is not used for the vertices of the diagonal between $t(v)$ and $t(u)$.

Lemma 4: *The dual graph of a triangulation is a tree, with each node of degree at most three.*

That each node has degree at most three is immediate from the fact that a triangle has at most three sides to share.

The nodes of degree one are leaves of T; nodes of degree two lie on paths of the tree; nodes of degree three are branch points.

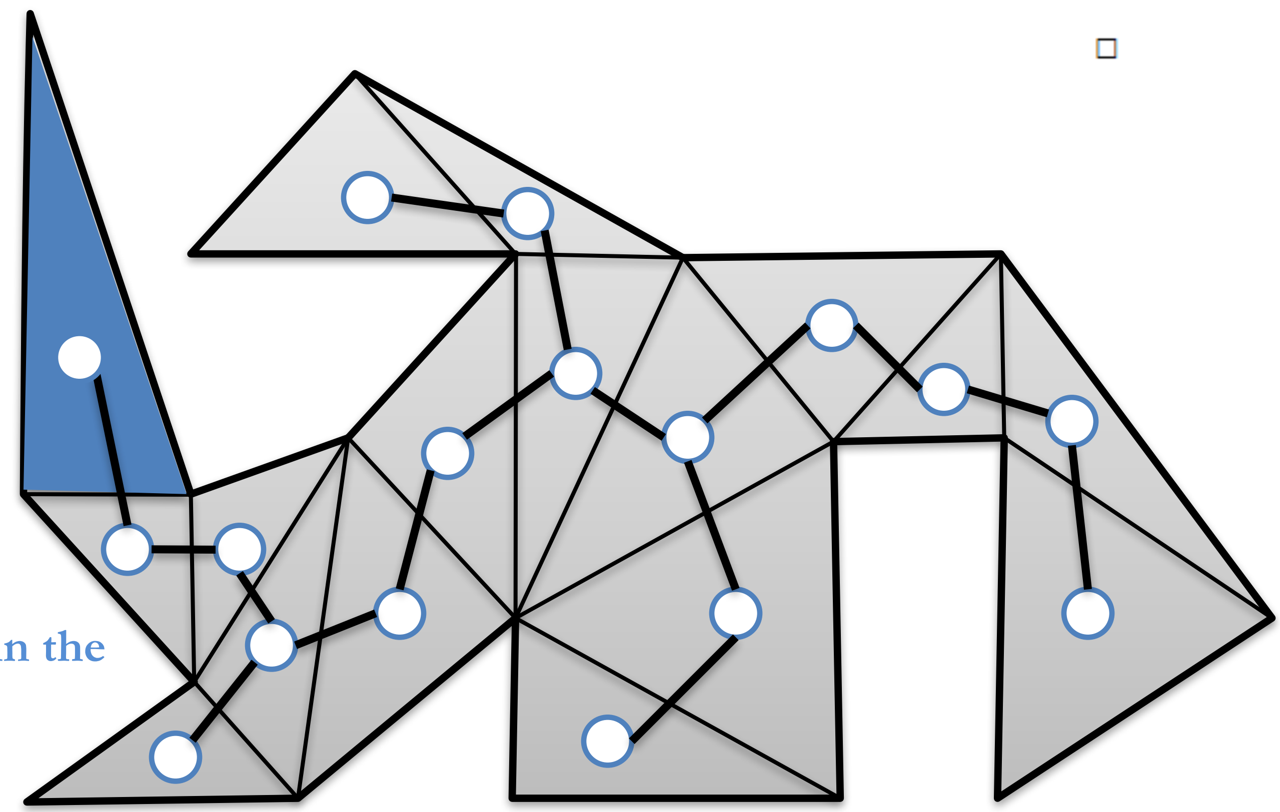
Three consecutive vertices of a polygon a, b, c form an *ear* of the polygon if ac is a diagonal; b is the ear *tip*. Two ears are *non-overlapping* if their triangle interiors are disjoint.

Theorem 3 (Meisters's Two Ears Theorem): *Every polygon of $n \geq 4$ vertices has at least two non-overlapping ears.*

Proof:

A leaf node in a triangulation dual corresponds to an ear. A tree of two or more nodes has $n-2 > 2$ nodes must have at least two leaves since every triangulation of a polygon $\mathcal{P}$ of n vertices consists of $n-2$ triangles.

$\square$



Example of an ear which corresponds to a leaf node in the dual graph of polygon triangulation

This theorem leads to an easy proof of the 3-colorability of triangulation graphs. The idea is to remove an ear for induction, which, because it only "interfaces" at its one diagonal, can be colored consistently.

Theorem 4 (3-coloring): *The triangulation graph of a polygon $\mathcal{P}$ may be 3-colored.*

Proof:

The proof is by induction on the number of vertices *n*. Clearly a triangle can be 3-colored.

Assume therefore that $n > 4$. Let the Theorem be true for any polygon with $m<n$ vertices, now does it hold for *n* vertices?

Since every polygon of $n > 4$ vertices has at least two non-overlapping ears (Theorem), thus $\mathcal{P}$ has an ear, let it be Δabc, with ear tip *b*. Form a new polygon $\mathcal{P}'$ by cutting off the ear: That is, replace the sequence *abc* in $\partial\mathcal{P}$ with *ac* in $\partial\mathcal{P}'$.

Now $\mathcal{P}'$ has $n - 1 < n$ vertices since it is missing only *b*. By the induction hypothesis $\mathcal{P}'$ can be *3-colored*, and by the induction base statement, the ear also can be *3-colored*. Now put the ear back, coloring *b* with the color not used at *a* and *c*. This is a 3-coloring of $\mathcal{P}$.

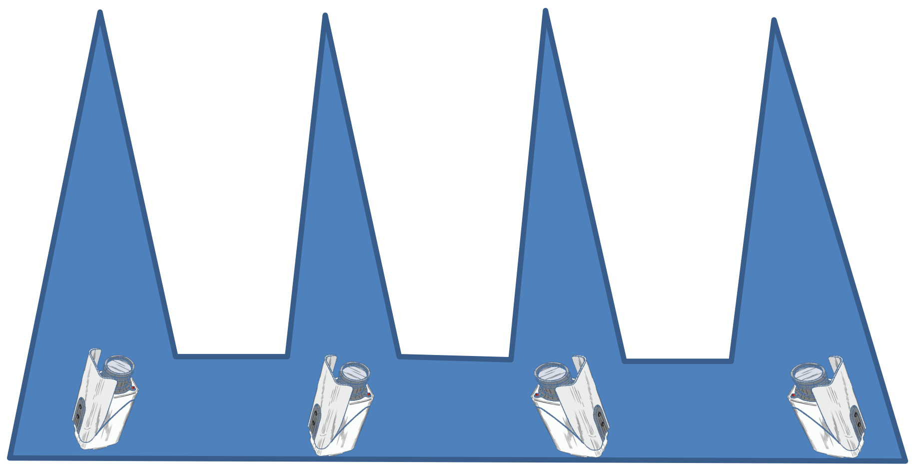
$\square$

Now we can conclude that a triangulated simple polygon can always be 3-colored. As a result, any simple polygon can be guarded with $\lfloor n/3 \rfloor$ cameras.

But perhaps we can do even better. Since a camera placed at a vertex may guard more than just the triangles it shares. Unfortunately, for any *n* there are simple polygons that require $\lfloor n/3 \rfloor$ cameras.

An example is a comb-shaped polygon with a long horizontal base edge and $\lfloor n/3 \rfloor$ prongs made of two edges each. The prongs are connected by horizontal edges. The construction can be made such that there is no position in the polygon from which a camera can look into two prongs of the comb simultaneously.

So we cannot hope for a strategy that always produces less than $\lfloor n/3 \rfloor$ cameras. In other words, the 3-coloring approach is optimal in the worst case. We just proved the Art Gallery Theorem.



Theorem 5: Art Gallery Theorem : *For a simple polygon with n vertices,* $\lfloor n/3 \rfloor$ *cameras/guards are occasionally and always sufficient to have every point in the polygon visible from at least one of the cameras/guards.*

Area of Polygon

In this section we will discuss one of the basic tools, area of a given polygon, which we will be extensively used to determine intersection between line segments, quantify visibility relations and ultimately lead to a triangulation algorithm.

Area of a Triangle

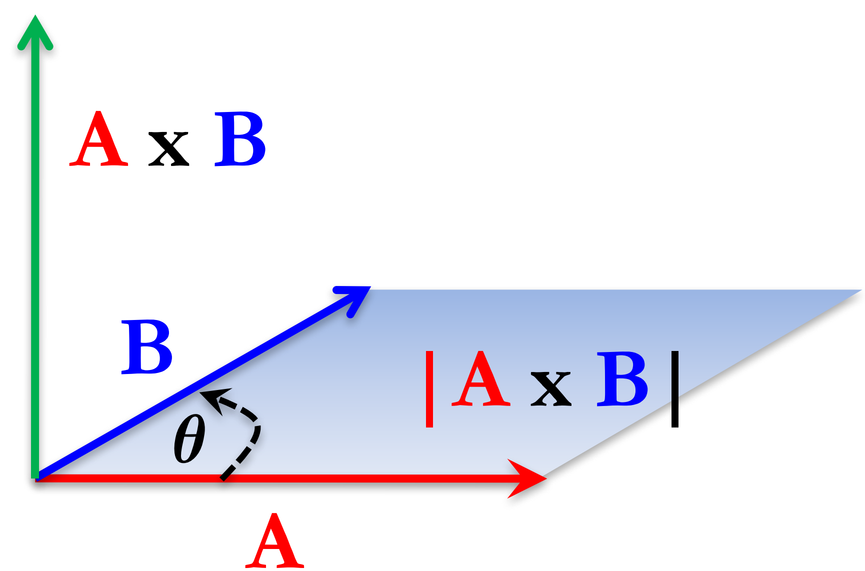
The area of a triangle t is one half the base times the altitude. However, this formula is cannot be directly used when we are given a triangle with arbitrary vertices a, b, c.

Let us denote this area by $\mathcal{A}(t)$, the length of base can be computed as $|a-b| \equiv \|a-b\|$, however the altitude cannot be directly computed from the available coordinates unless the triangle happens to be oriented with one side parallel to one of the axes.

Cross Product

From linear algebra we know that the magnitude of the cross product of two vectors is the area of the parallelogram they determine. If A and B are vectors, then $|A \times B|$ is the area of the parallelogram with sides A and B, as shown in the following figure.

The magnitude of the cross product of two vectors is the area of the parallelogram they determine



Since any triangle can be viewed as half of a parallelogram, this gives an immediate method of computing the area from vertices coordinates.

Just let $A = b - a$ and $B = c - a$. Then the area is half the length of $A \times B$.

The cross product can be computed from the following determinant, where *i*, *j* and *k* are unit vectors in the *x*, *y*, and *z* directions respectively:

$$\begin{vmatrix} i & j & k \\ A_0 & A_1 & A_2 \\ B_0 & B_1 & B_2 \end{vmatrix} = (A_1B_2 - A_2B_1)i - (A_2B_0 - A_0B_2)j + (A_0B_1 - A_1B_0)k$$

For 2-dimensional space, $A_2 = B_2 = 0$, hence this reduces to $(A_0B_1 - A_1B_0)k$. The cross product is a vector normal/perpendicular to the plane containing the triangle, thus the area can be given by,

$$\mathcal{A}(t) = \frac{1}{2}(A_0B_1 - A_1B_0)$$

Since $A = b - a$ and $B = c - a$, this will lead to,

$$\begin{aligned} 2\mathcal{A}(t) &= \big((b_0 - a_0)(c_1 - a_1) - (b_1 - a_1)(c_0 - a_0)\big) \\ &= a_0b_1 - a_1b_0 + a_1c_0 - a_0c_1 + b_0c_1 - c_0b_1 \end{aligned}$$

Hence we have expressed the area of a given triangle in terms of the coordinates of its vertices.

Determinant Form

There is another way to represent the calculation of the cross product which is formally identical but generalizes more easily to higher dimensions.

The expression obtained before is the value of the 3 x 3 determinant of the three point coordinates, with the third coordinate replaced by 1.

Lemma 5: *Twice the area of a triangle* $t = (a, b, c)$ *is given by,*

$$2\mathcal{A}(t) = \begin{vmatrix} a_0 & a_1 & 1 \\ b_0 & b_1 & 1 \\ c_0 & c_1 & 1 \end{vmatrix} = (b_0 - a_0)(c_1 - a_1) - (c_0 - a_0)(b_1 - a_1)$$

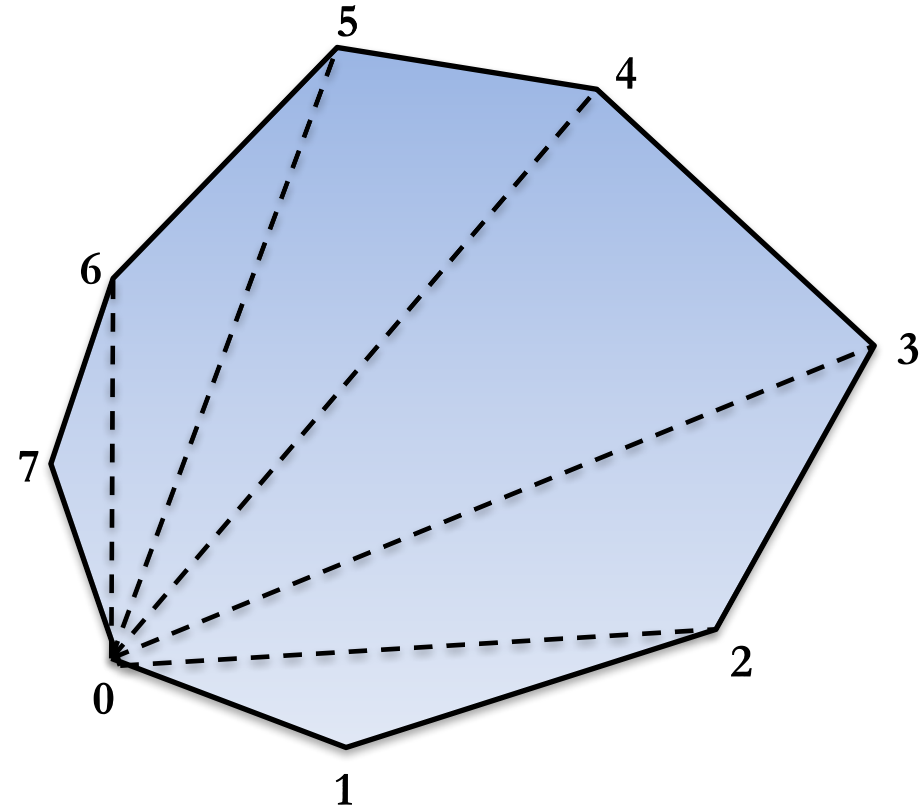
Area of a Convex Polygon

- Now that we have an expression for the area of a triangle, it is easy to find the area of any polygon by first triangulating it, and then summing the triangle areas.

- However it would be easier if we can avoid this complicated step to just compute the area of a polygon.

- Let's first consider convex polygons where triangulation is a trivial task.

- Every convex polygon may be triangulated as a "fan," with all diagonals share a common vertex which is denoted by the fan center, any vertex in a convex polygon can serve as the fan center.



A convex polygon can be triangulated as a fan with all diagonals share a common vertex known as the fan center

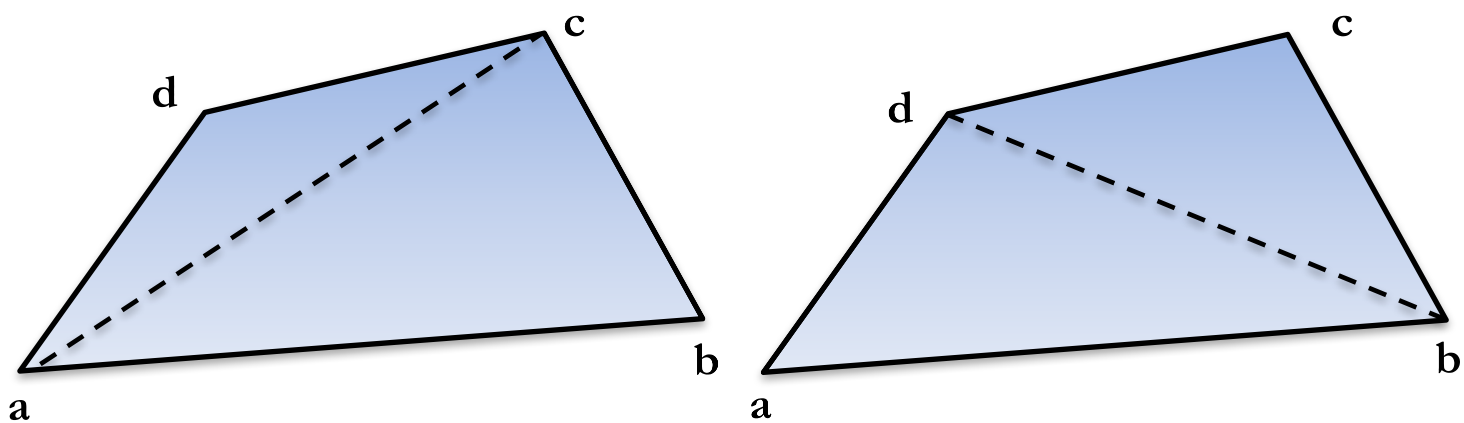
Therefore the area of a polygon with vertices $v_0, v_1, \dots, v_{n-1}$ labeled counterclockwise, with v_0 as the fan center can be calculated as

$$\mathcal{A}(\mathcal{P}) = \mathcal{A}(v_0, v_1, v_2) + \mathcal{A}(v_0, v_2, v_3) + \cdots + \mathcal{A}(v_0, v_{n-2}, v_{n-1})$$

Area of a Convex Quadrilateral

The area of a convex quadrilateral $Q = (a, b, c, d)$ may be written in two ways, depending on the two different triangulations shown in the following figure:

$$\mathcal{A}(Q) = \mathcal{A}(a,b,c) + \mathcal{A}(a,c,d) = \mathcal{A}(d,a,b) + \mathcal{A}(d,b,c)$$



Two possible triangulations of a convex quadrilateral

Using the cross-product formula for area of a triangle, the area of a convex quadrilateral can be written as follows;

$$2\mathcal{A}(Q) = (a_0b_1 - a_1b_0 + a_1c_0 - a_0c_1 + b_0c_1 - c_0b_1) \\ + (a_0c_1 - a_1c_0 + a_1d_0 - a_0d_1 + c_0d_1 - c_1d_0)$$

Note that $a_1c_0 - a_0c_1$ appear in $\mathcal{A}(a,b,c)$ and $\mathcal{A}(a,c,d)$ with opposite signs, hence they cancel, thus the terms corresponding to the diagonal ac cancel. Likewise, the terms corresponding to the diaongal bd in the second triangulation cancel. Hence the expression of the area will be the same regardless of the triangulation being used. The following theorem generalizes the area expression in terms of polygon vertices.

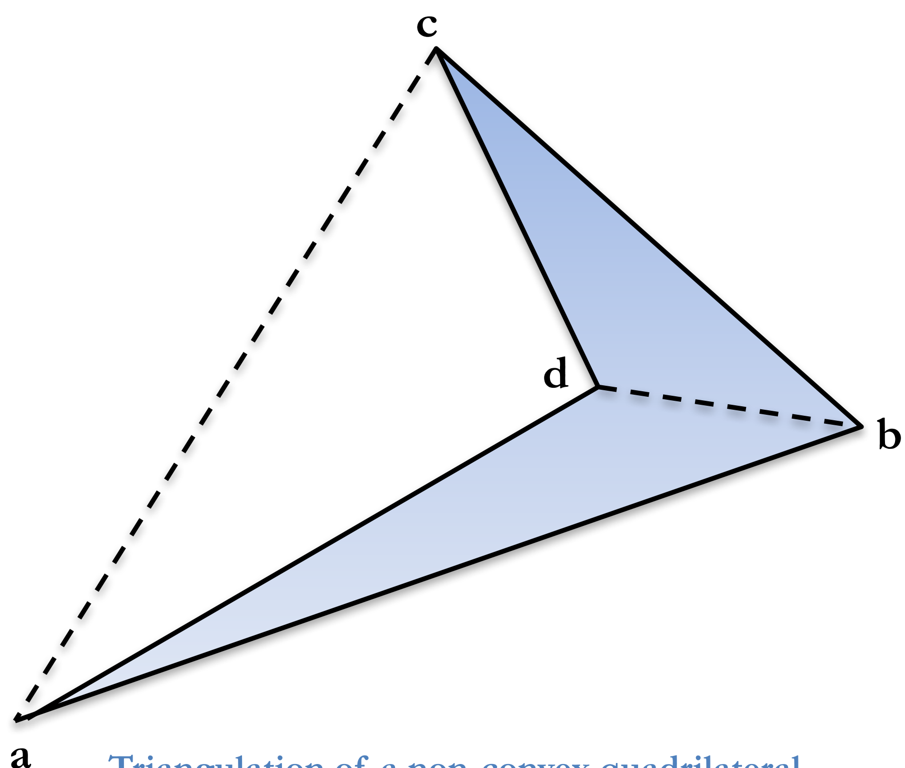
Theorem 6: *Let $\mathcal{P}$ be a convex polygon with vertices $v_0, v_1, \dots, v_{n-1}$ labeled counterclockwise, let the coordinates of the i-th vertex v_i be denoted as x_i and y_i, hence twice the area of the convex polygon $\mathcal{P}$ is given by,*

$$2\mathcal{A}(\mathcal{P}) = \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i)$$

Area of a Non-convex Quadrilateral

Now suppose we have a non-convex quadrilateral $Q = (a, b, c, d)$. Then there is only one triangulation, using the diagonal db. See the following figure.

But we just showed that the algebraic expression obtained is independent of the diagonal chosen, so it must be the case that the equation $\mathcal{A}(Q) = \mathcal{A}(a, b, c) + \mathcal{A}(a, c, d)$ is still true even through the diagonal ac which is exterior to Q, this has an obvious interpretation, which is that $\mathcal{A}(a, c, d)$ should be negative and hence is subtracted from the bigger triangle Δabc. Note that (a, c, d) is a clockwise path, hence from the cross product formulation, this triangle should have a negative area.



Triangulation of a non-convex quadrilateral

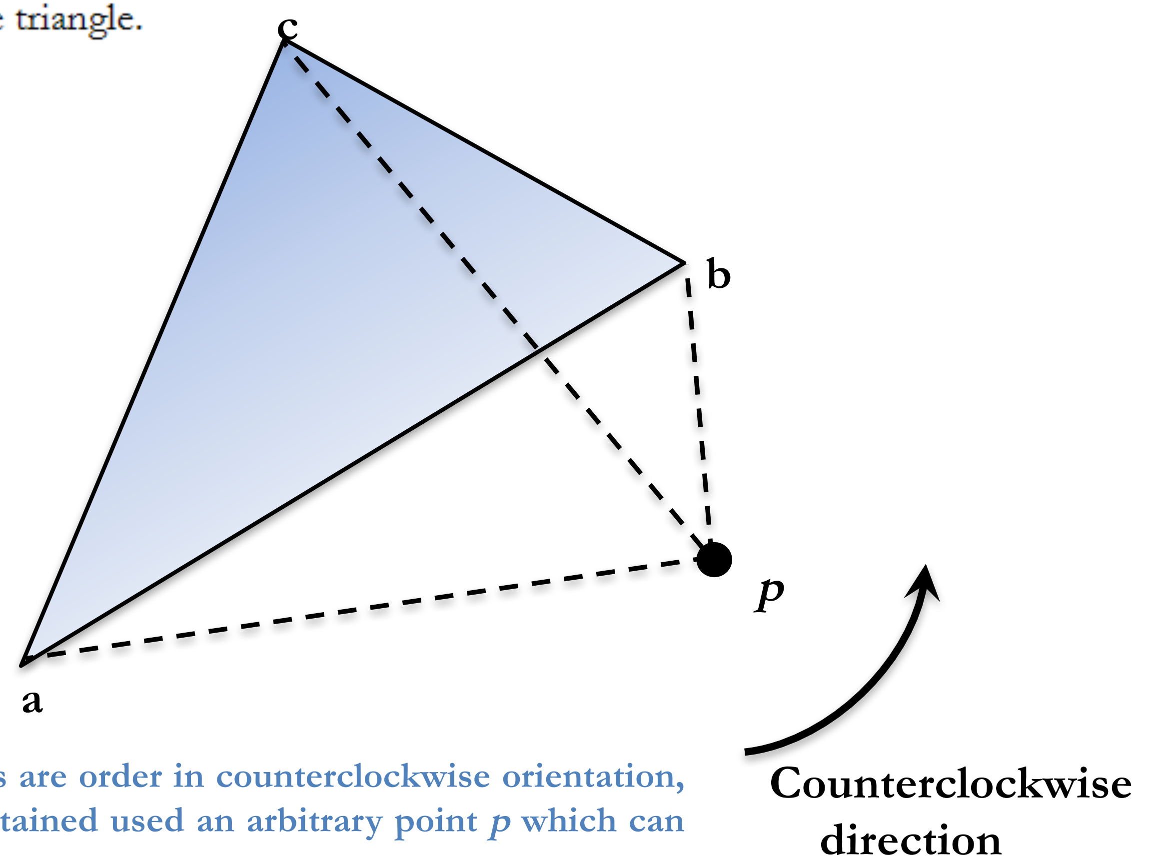
Area of Non-convex Polygon

In this subsection, we will formalize our observations to obtain the area of a general non-convex polygons. Let us first derive a general form of summing the areas of the triangles in a given triangulation using areas based on an arbitrary, may be external, point p.

Let $t = \Delta abc$ be a triangle with vertices ordered in a counterclockwise manner, let p be any points in the plane, then we claim that:

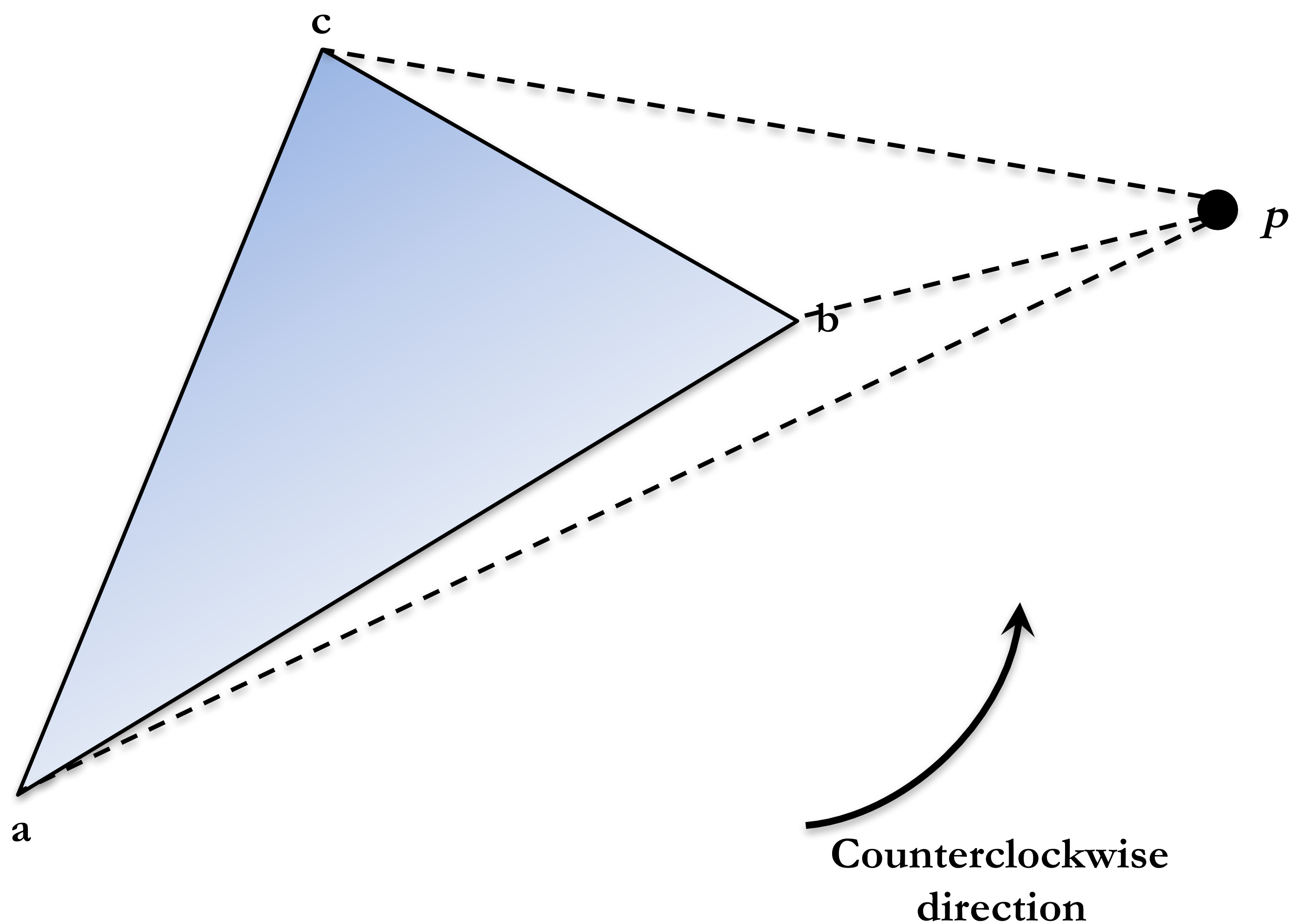
$$\mathcal{A}(t) = \mathcal{A}(p,a,b) + \mathcal{A}(p,b,c) + \mathcal{A}(p,c,a)$$

Consider the figure, the first term $\mathcal{A}(p,a,b)$ is negative because the vertices are clockwise, on the other hand the remaining two terms are positive because the vertices are counterclockwise. Hence the term $\mathcal{A}(p,a,b)$ subtracts the portion of the quadrilateral (p,b,c,a) which lies outside the triangle t leaving only the area of the triangle.



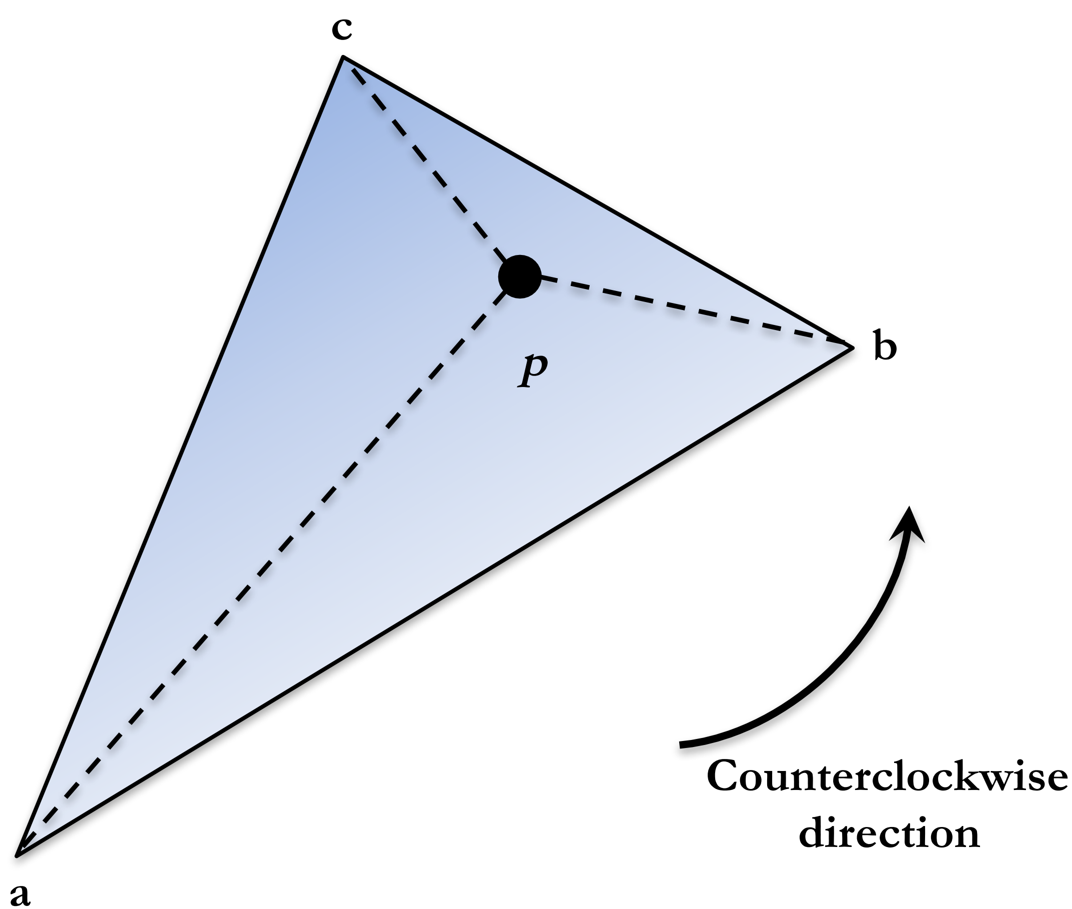
Consider a triangle whose vertices are order in counterclockwise orientation, the area of the triangle can be obtained used an arbitrary point p which can lie outside that triangle.

Consider another location for this external point (see the figure above), both $\mathcal{A}(p,a,b)$ and $\mathcal{A}(p,b,c)$ are negative because the vertices are clock wise, and they are removed from $\mathcal{A}(p,c,a)$ which is positive.



Considering another location for the external point p

All other positions for p in the place exterior to the triangle t is equivalent to either the first or the second case, the equation also holds when p is internal where $\mathcal{A}(p, a, b), \mathcal{A}(p, b, c)$ and $\mathcal{A}(p, c, a)$ are all positive.



Therefore we have established the following lemma.

Lemma 6: *If* $t = \Delta abc$ *is a triangle, with vertices oriented counterclockwise and* p *is any point in the plane, then*

$$\mathcal{A}(t) = \mathcal{A}(p, a, b) + \mathcal{A}(p, b, c) + \mathcal{A}(p, c, a)$$

The following theorem generalize the preceding lemma to obtain a generalized equation for arbitrary polygons.

Theorem 7 - Area of Polygon: *Let* $\mathcal{P}$ *be a simple polygon (convex or nonconvex), having vertices* $v_0, v_1, \dots, v_{n-1}$ *labeled counterclockwise, and let* p *be any point in the plane, then*

$$\mathcal{A}(\mathcal{P}) = \mathcal{A}(p, v_0, v_1) + \mathcal{A}(p, v_1, v_2) + \mathcal{A}(p, v_2, v_3) + \cdots + \mathcal{A}(p, v_{n-2}, v_{n-1}) + \mathcal{A}(p, v_{n-1}, v_0)$$

Let the coordinates of the i-th vertex v_i *be denoted as* x_i *and* y_i*, hence twice the area of the polygon* $\mathcal{P}$ *is given by,*

$$2\mathcal{A}(\mathcal{P}) = \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i) = \sum_{i=0}^{n-1} (x_i + x_{i+1})(y_{i+1} - y_i)$$

Proof left as homework .. ☹

Segment Intersection

We still have one further step to be able to develop an algorithm to triangulate a given polygon, in this subsection we will discuss how can we detect an intersection between two given segments.

Diagonals

The key step to triangulate a polygon is to find a diagonal of that polygon, which is a direct line of sight between two vertices v_i and v_j. The segment $v_i v_j$ will not be a diagonal if it is blocked by a portion of the polygon's boundary, that is it intersects the boundary $\partial\mathcal{P}$ at any point rather than v_i and v_j.

Hence the segment $v_i v_j$ is considered to be blocked if it intersects an edge of the polygon. Note that if $v_i v_j$ only intersects an edge e at its endpoint, i.e. at one vertex of the polygon, perhaps only a grazing contact with the boundary, it is still effectively blocked, because diagonals must have clear visibility.

Thus we can re-define a diagonal as follows,

Definition 11: *The segment* $s = v_i v_j$ *is a* **diagonal** *of the polygon* $\mathcal{P}$ *if and only if:*

(1) For all edges e *of* $\mathcal{P}$ *that are not incident to either* v_i *or* v_j, s *and* e *do not intersect, i.e.* $s \cap e = \emptyset$
(2) s *is internal to* $\mathcal{P}$ *in a neighborhood of* v_i *and* v_j.

Condition (1) of this definition has been phrased such that the *diagonalhood* of a segment can be determined without finding the actual point of intersection between s and e, hence it is enough to detect the intersection. Recall the original definition of the diagonal which states that *a diagonal only intersects polygon edges at the diagonal endpoints*, using this phrasing would require the computation of the intersection points and comparing them to the endpoints. The purpose of condition (2) is to distinguish internal from external diagonals, as well as to rule out collinear overlap with an incident edge. In the following subsections we will be concentrating on detecting intersection between two given segments.

Problems with Slopes

Let $s = v_i v_j = ab$ and $e = cd$. Now we want to determine whether the two segments ab and cd do intersect or not.

The first attempt towards this goal is to find the point of intersection between the lines L_1 and L_2 containing these segments, this can be achieved by solving the two linear equations in slope-intercept form and then checking if the point falls on the segments.

Although this method might work and not difficult to code, it is error prone.

There are two special cases to handles, a vertical segment whose line's slope is infinite and parallel segments whose lines do not intersect. In addition checking that the point of intersection falls on the segments can lead to numerical precision problems, hence we will avoid slopes altogether.

The Left Predicate

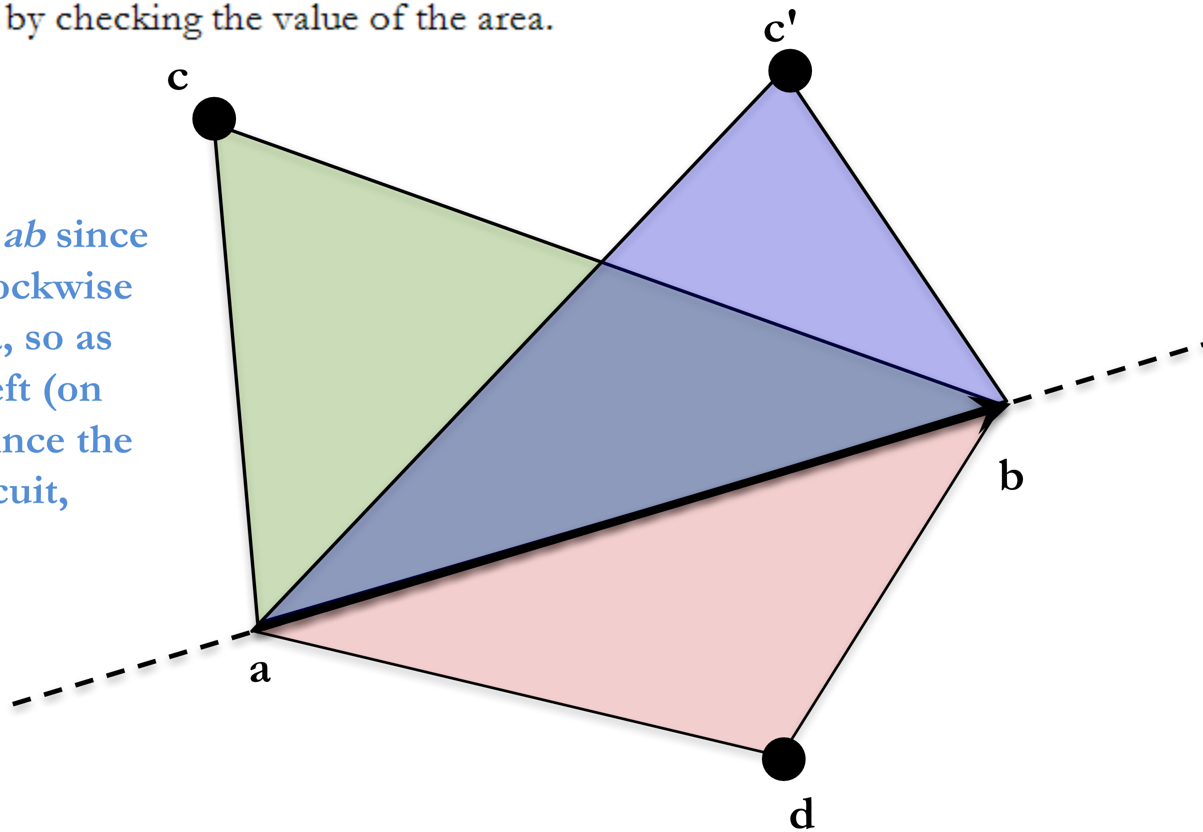
Checking whether two segments intersect can be established by determining whether or not a point is to the left of a directed line.

This is called the Left predicate, in the next subsection we will discuss how to use such predicate to check for segment intersection, here we will concentrate on the Left predicate itself.

Two points given a particular order (a, b) determine a directed line moving from the first point a to the second point b. If another point c is to the left of this directed line, then the triple (a, b, c) will form a counterclockwise circuit, hence the triangle constructed by such circuit, i.e. $t = \Delta abc$, will have a positive area. This motivates the following lemma

Lemma 7: *Let L be a directed line determined by two points given in a particular order* (a, b), *let* c *be a point,* c *is said to be to the left of* (a, b) *if and only if the area of the counterclockwise triangle,* $t = \Delta abc$ *is positive, that is* $\mathcal{A}(t) > 0$.

However, what happens if c is collinear with ab? Then the determined triangle will have zero area, thus we can also detect collinearity by checking the value of the area.



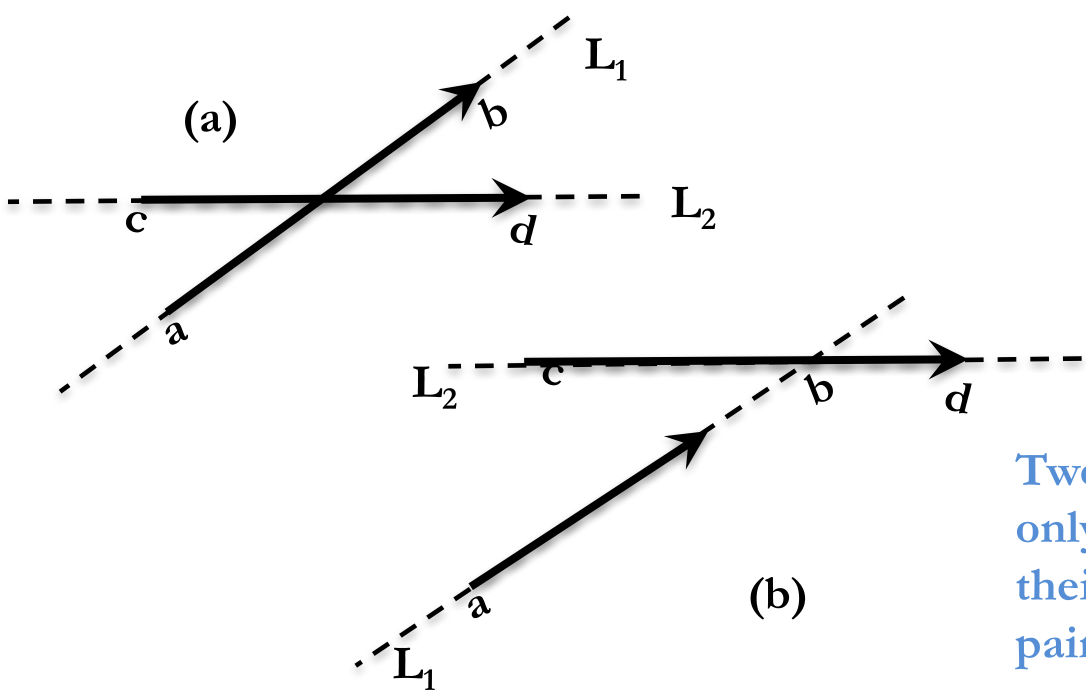
c is on the left of the directed line *ab* since the triangle *abc* forms a counterclockwise circuit, hence it has a positive area, so as the point *c'* while *d* is not on the left (on the right) of the directed line *ab* since the triangle *abd* forms a clockwise circuit, hence its area is negative

Intersection Detection

Intuitively, if two segments ab and cd intersect in their interiors, that is at a point belong to both segments, then c and d are split by the line L_1 containing the segment ab, that is c is to one side and d to the other.

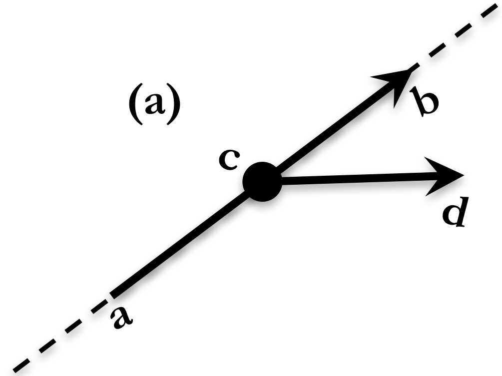
And likewise, a and b are split by the line L_2 containing the segment cd, that is a is to one side and b to the other.

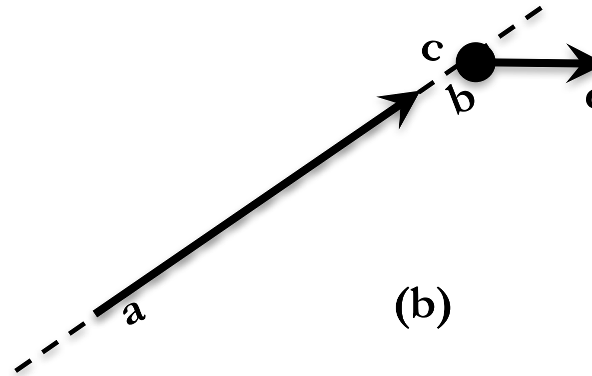
Note that neither of these conditions alone is sufficient to guarantee intersection, however we should make sure first that we do not have the case where three of the four endpoints are collinear. This is referred to as *proper intersection* where we force non-collinearity when two segments intersect at a point interior to both.



Two segments intersect (a) if and only if their endpoints are split by their determined lines, (b) both pair of endpoints must be split

- Now we should deal with the special case of *improper intersection* between two segments, this occurs when an endpoint of one segment lies somewhere on the other segment, this can only happen if there points are collinear, however collinearity is not a sufficient condition.





To check for improper intersection between two segments where an endpoint of one segment lies somewhere on the other segment, collinearity is not a sufficient condition, in (a) and (b) a,c,b are collinear however (a) has improper intersection while (b) not.

- Hence what we need is to decide if an endpoint of a segment lies between the endpoints of the other segment.

- We would like to compute the "betweenness" predicate without using slopes.

- If the point c is known to be collinear with a and b, the betweenness check can proceed as follows;

  – If ab is not vertical, then c lies on ab if and only if the x coordinate of c falls in the interval determined by the x coordinates of a and b. If ab is vertical, then a similar check on y coordinates determines betweenness.

Segment Intersection

- Now, we can determine whether two segments intersect or not using the following condition:

Corollary 2: *Two segments intersect if and only if they intersect properly or one endpoint of one segment lies between the two endpoints of the other segment.*

Triangulation

Finding a Diagonal

In order to perform polygon triangulation, we need first to know how to find a diagonal of the given polygon.

Recall that diagonals are characterized by two main conditions: (1) non-intersection with polygon edges and (2) being interior to the polygon.

If we ignore the second condition, finding a diagonal will be straightforward: Consider a potential diagonal s connecting between a pair of polygon vertices v_i and v_j, for every edge e of the polygon $\mathcal{P}$ not incident to either v_i or v_j, check if e intersect s, as soon as an intersection is detected, s will be declared not to be a diagonal, if no such edge intersects s, then s might be a diagonal, since we have already ignored the second condition, we should check whether it is interior or exterior to the polygon, we will investigate this issue in the next subsection.

InCone – Interior/Exterior Diagonal

It is time now to distinguish between internal and exterior diagonals, and at the same time handle the case when one or more polygon edges are incident to the diagonal endpoints (improper intersection).

Let ab be a potential diagonal which satisfies condition (1), that is it does not intersect with any polygon edge.

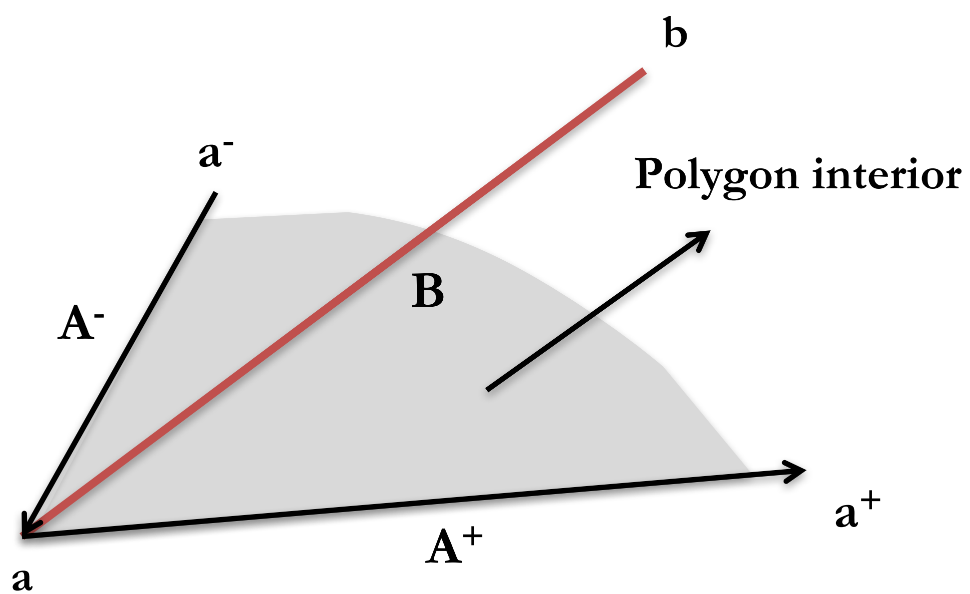
Let a^+ and a^- be the two neighboring vertices to a which are directly connected to a via one edge.

Let B be a vector which lies along the diagonal ab. Let A^+ and A^- be vectors lying along the two consecutive edges of the polygon aa^+ and aa^- respectively.

The InCone predicate determines if the vector B lies strictly in the open cone counterclockwise between two other vectors A^+ and A^-. Such a procedure will suffice to determine diagonals, as will be detailed below. For the moment we concentrate on designing InCone.

This would be a straightforward task if the apex, i.e. a, of the cone is a convex angle; It is clear from the following figure that ab is internal to $\mathcal{P}$ if and only if it is internal to the cone whose apex is a, and whose sides pass through a^+ and a^-.

This can be easily determined via our Left predicate: a^- must be left of ab, and a^+ must be left of ba. Both left-ofs should be strict for ab to exclude collinear overlap with the cone boundaries. Hence the condition will be Left(ab,a$^-$) && Left(ba,a$^+$)



Diagonal ab is in the cone determine by a^+ and a^- which are the two neighboring vertices to a which are directly connected to a via one edge. When the cone apex a is convex, a^+ should lie to the left of ab and a^- should lie to the left of ba.

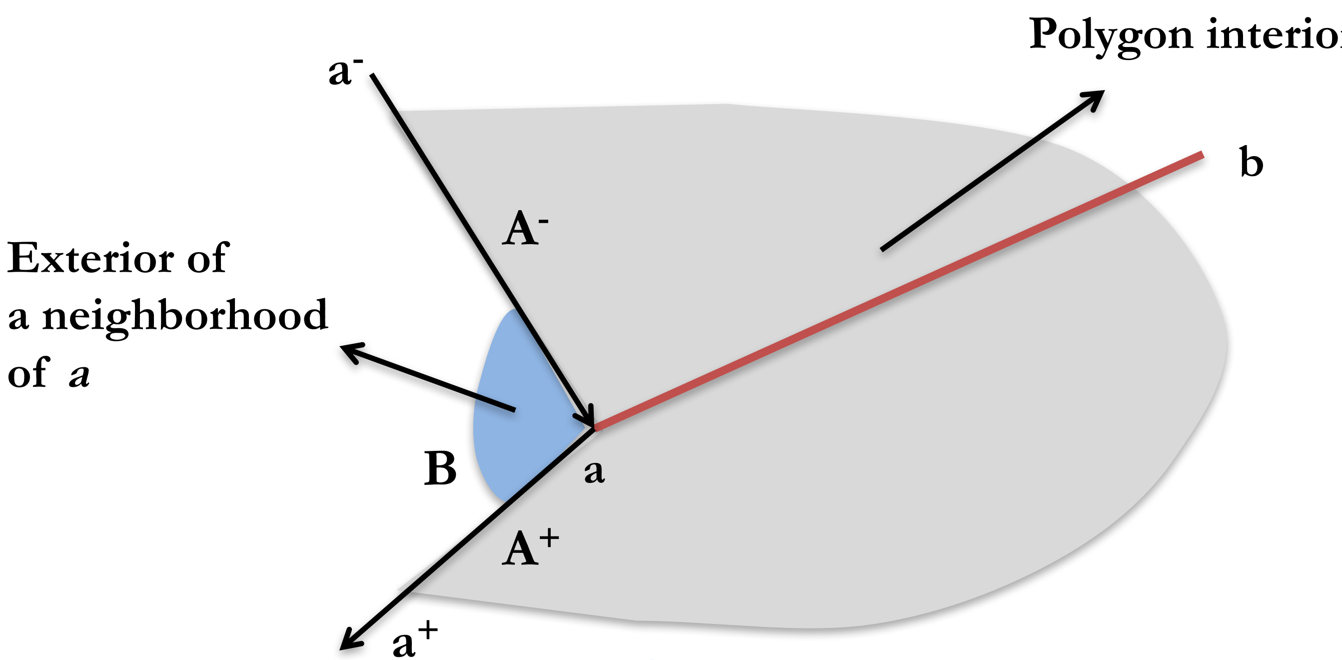
However, the following figure shows that these conditions do not suffice to characterize internal diagonals when a is reflex: a^+ and a^- could be both left of, or both right of, or one could be left and the other right of, an internal diagonal.

But note that the *exterior* of a neighborhood of *a* is now a cone as in the convex case.

So it is easiest in this case to characterize ab as internal if an only if it is not external.

It is not the case that both *a+* is left or on *ab* and *a-* is left or on *ba*.

Note that this time the left-ofs must be improper, permitting collinearity, as we are rejecting diagonals that satisfy these conditions. Hence the condition will be !(Left(ab,a^+) && Left(ba,a^-)).



Cone apex a is reflex, hence the left-of conditions from a^+ and a^- are not sufficient to determine internal diagonals.

Finally, distinguishing between the convex and reflex cases is easily accomplished with one invocation of Left: a is convex iff a- is left or on aa^+. Note that if $(a-, a, a^+)$ are collinear, the internal angle at a is π, which we defined as convex.

Now we can determine if ab is a diagonal:

This is true if and only if ab does not intersect any of the polygon edges and ab is InCone and ba is InCone too.

The last two conditions are to make sure that ab is internal and to cover the edges incident to the endpoints and not examined in the first condition.

To efficiently implement this check, the InCones should be checked first, because they are each constant-time calculations, performed in the neighborhood of a and b without regard to the remainder of the polygon, whereas the first condition includes a loop over all n polygon edges.

If either InCone call returns false, the first condition will not be checked which will save computational overhead.

Triangulation by Ear Removal

We are now ready to outline an algorithm for polygon triangulation. One possible method is to mimic the proof of the triangulation theorem: Find a diagonal, cut the polygon into two pieces and do the same for each piece till we end up with bunch of triangles.

Such method is called *diagonal-based algorithm*, however this method results in rather inefficient code.

Diagonal-based triangulation is an $O(n^4)$ algorithm, there are $\binom{n}{2} = O(n^2)$ diagonal candidates and testing each for diagonalhood costs $O(n)$, repeating this $O(n^3)$ computation for each of the $n-3$ diagonals leads to $O(n^4)$.

We can speed this up by a factor of n by exploiting the two ears theorem: not only do we know there must be an internal diagonal, we know there must be an internal diagonal that separates off an ear.

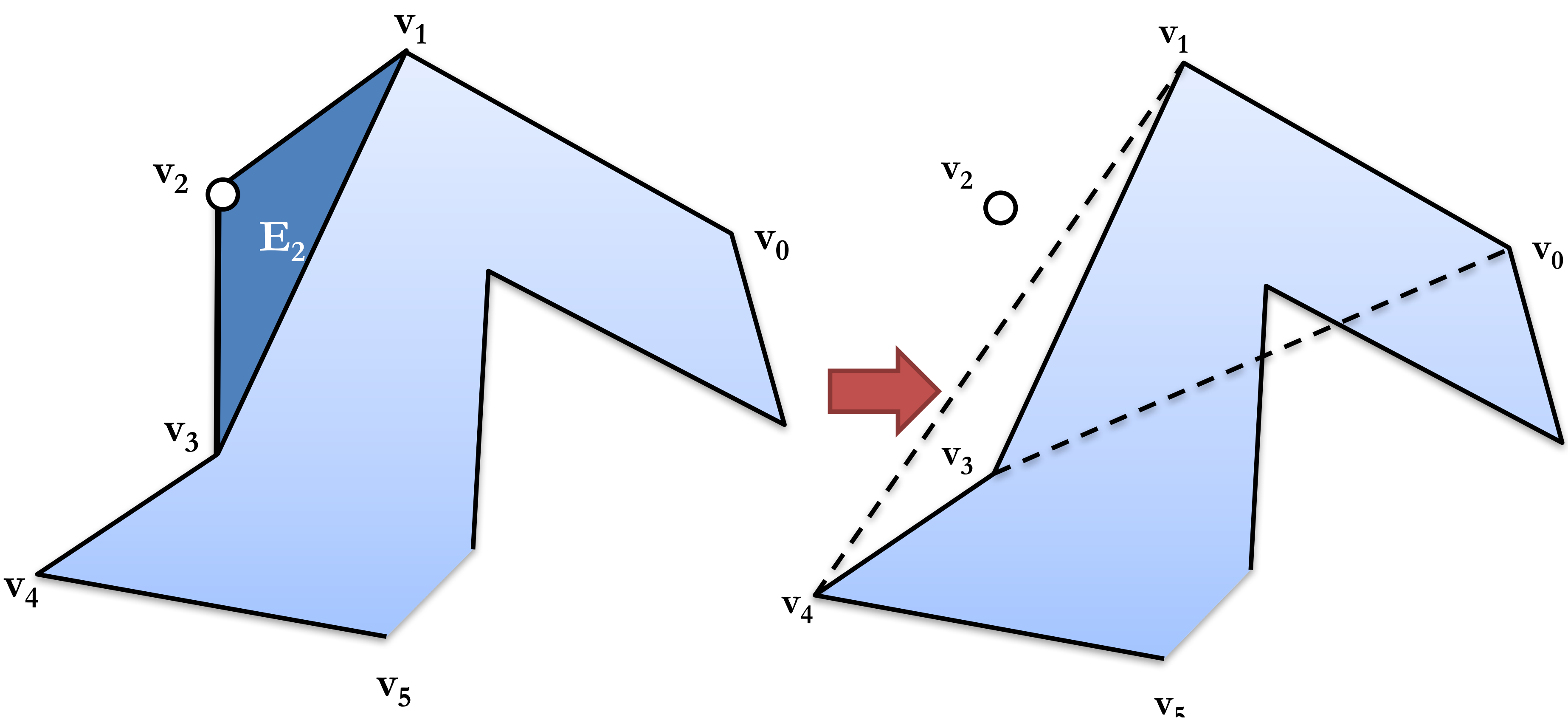
There are only $O(n)$ *ear diagonals* candidates, which connect two consecutive vertices $(v_i, v_{i+2}),\ i = 0,1,\dots,n-1$.

This also makes the recursion easier, since after removing the ear we will have an ear which is already triangulated and the rest of the polygon, previously we had two sub-polygons to recurse on in order to be triangulated. Thus we can achieve a worst-case complexity of $O(n^3)$ using this way.

We can further improve this algorithm, the key idea which permits improvement here is that removal of one ear does not change the polygon very much, in particular, this does not change whether or not many of its vertices are potential ear tips. This suggests first determining for each vertex v_i, whether it is a potential ear tip in the sense that (v_{i-1}, v_{i+1}) is a diagonal. This uses $O(n^2)$, however this expensive step need not be repeated.

Let v_0, v_1, v_2, v_4, v_5 be five consecutive vertices of $\mathcal{P}$, and suppose that v_2 is an ear tip and the ear $E_2 = \Delta(v_1, v_2, v_3)$ is deleted. Which vertices' status as ear tips might change? Only v_1 and v_3.

Consider v_4, for example. Whether it is an ear tip depends on whether v_3v_5 is a diagonal. The removal of E_2 leaves the endpoints of segment v_3v_5 unchanged. Hence the status of v_4 is not changed by ear removal, this is same for v_5.



Clipping an ear $E_2 = \Delta(v_1, v_2, v_3)$, here the ear status of v_1 and v_3 change from true to false.

Thus the algorithm can be formulated as follows;

Algorithm: Triangulate via ear removal

(1) Initialize the ear tip status of each vertex
(2) While $n > 3$ do
    a. Locate an ear tip v_2 where $(v_{1,} v_3)$ is a diagonal.
    b. Update the ear tip status of v_1 and v_3 where v_1 is an ear if $v_0 v_3$ is a diagonal and v_3 is an ear if $v_1 v_4$ is a diagonal.
    c. Cut of the ear v_2.

Thank You