An Upper Bound on Trilaterating Simple Polygons

Matthew Dippel*   Ravi Sundaram †

Abstract

In this work, we introduce the Minimum Trilateration Problem, the problem of placing distance measuring guards in a polygon in order to locate points in the interior. We provide the first non-trivial bounds on trilaterating simple polygons, by showing that $\lfloor \frac{8N}{9} \rfloor$ guards suffice for any non-degenerate polygon of N sides, and present an $O(N \log N)$ algorithm for the corresponding placement. We also show how this mapping can be efficiently inverted, in order to determine a point's location given its distances to the guards which can see it.

1 Introduction

Trilateration is the technique of determining absolute locations of points using distances and the geometry of circles and spheres. For example, in 2-D if the two distances of a point from two fixed centers are known then there are only two possible candidate locations for the point. If three distances of a point from three fixed, non collinear points are known, there is only one possible location for the unknown point.

Formally, suppose we have a set of known points $\{r_i\}$, and their corresponding distances $\{d_i\}$ to an unknown point p. Then, to trilaterate p, we must solve the system:

$$\|r_i - p\| = d_i \tag{1}$$

for all possible solutions p. If there exists a unique solution, then we say that the set $\{r_i\}$ trilaterates p.

Observation 1 *If r_1, r_2, r_3 are non collinear points, and the d_i's are valid distances, then the system in Equation 1 is always solvable for a unique p.*

We define distance measuring guards as points which can measure the distance to other points in their visibility region. Given a polygon P, we wish to find a set of distance measuring guards R such that for every point p, when we consider only those $\{r_i\}$ which can view p, the system in Equation 1 has a unique solution in P. We will formalize this problem in Section 2.

We call the problem of finding the smallest such guard set for a polygon as the **Minimum Trilateration Problem**. Our results show upper bounds for this problem analogous to the upper bounds for the Art Gallery Problem [12]. As the Art Gallery Problem is analogous to dominating sets in visibility graphs, our problem also has an analogous graph theory problem known as the metric dimension of the graph. [8]. There has also been work on guarding polygons where each point must be viewed by multiple guards [3], but although they mention trilateration as an application, k-guarding a polygon is not sufficient for unambiguous trilateration.

First, we show that when constrained to certain properties, partitioning a polygon and finding a trilaterating guard set for each individual piece can result in a valid trilateration of the original polygon. We demonstrate that every polygon admits such a partition, and show how to efficiently map these partitions into trilaterating guard sets of size no more than $8N/9$ when the polygon has N sides.

We also show how these algorithms and bounds can be extended to the case where the polygon is not in general position. This is an important case because of the role that collinearity plays in our problem. We show that even in this case, collinear guards can locate points by taking advantage of the visibility geometry of the polygon.

2 Simple Trilateration

2.1 Definitions

Let P be a simple polygon, which may or may not be in general position (both cases will be addressed in this paper). For any two points $a, b \in P$, we say that a and b are mutually visible if $\overline{ab} \subset P$. For a specific point p, we define the visibility region $V(p)$, as the set of all points visible from p. The kernel of P, $K(P)$ is the set of points $k \in P$ such that $V(k) = P$. A polygon is star-shaped if $K(P)$ is non-empty.

For a specific point r, we can define a "vision-masked distance" function, $d_r : P \to \mathbb{R}$, such that $d_r(p) = \|p - r\|$ if r can view p, and -1 otherwise. For

*College of Computer Science, Northeastern University, mdippel@ccs.neu.edu

†College of Computer Science, Northeastern University, koods@ccs.neu.edu

a group of k points R, we similarly define $D_R : P \to \mathbb{R}^k$ as the vector of vision-masked distances from p to the various $r \in R$.

Our goal is to find a guard set such that, for all pairs of points $p, q \in P$, we have $D_R(p) \neq D_R(q)$. We formalize this with the following definition:

Definition 1 *For a simple polygon P, and a finite set of points $R \subset P$, we say that R trilaterates P without ambiguity if the vision-masked distance vector function $D_R(p)$ is injective over the domain P.*

2.2 Simple cases

There are several simple cases to consider. The most obvious is when P is the unbounded plane, and R is a set of three, non-collinear guards. Then, by solving a system of equations describing the intersection of three circles, we can uniquely locate any point p. If P is a star shaped polygon with at least 3 non-collinear points in the kernel, then we can trilaterate P with these three points.

When the kernel of P includes two points on the same edge, we can trilaterate P with only those two points. To see why, consider when R is two such points a and b, and we attempt to solve System 1. Solving the system of equations yields two points, p and p', reflected across $\overline{ab}$. WLOG p is on the same side of $\overline{ab}$ as the polygon P. Then, since a and b are in the kernel of P and are on an edge of it, this edge alone blocks their view of the other half of the plane induced by $\overline{ab}$. Thus we know that p' is not in the domain polygon P, and return p as the correct point. We give a generalized observation:

Observation 2 *Let R be a guard set in P, and consider two $a, b \in R$. If $V(a) \cap V(b)$ is entirely on one side of the closed half plane induced by $\overline{ab}$, then a and b can trilaterate points in the set $V(a) \cap V(b)$.*

The logic is the same as above. Solving System 1 with the distances to a and b yields two points, in opposite half planes induced by $\overline{ab}$. Thus we can rule out one of them from our domain, based on which one is on the same side of $\overline{ab}$ as the region $V(a) \cap V(b)$.

2.3 Partition and cover

One possible approach for finding a trilaterating set is to partition P into polygons which are simple to trilaterate (star-shaped or otherwise), individually trilaterate each one, and take the union of all guard sets. However, this doesn't always work. Consider the simple case where we wish to trilaterate a square, and we split it into two triangles via a diagonal. If we trilaterate each piece with two guards on the shared diagonal, the square cannot be trilaterated, as all guards are collinear. If we partition the polygon and use two guards on an edge to cover a piece, we need to be sure that the piece sharing that edge does not also put guards on it. We address this issue with the following lemma:

Lemma 1 *Let P be a simple polygon, partitioned into $P_1, P_2, ..., P_k$. Let $R_i \subset K(P_i)$ be sets of disjoint guards that trilaterate each respective polygon in the partition. Then, if $R_i \cup R_j$ contains three non-collinear guards for all i, j, R trilaterates P.*

Proof. We show that given any $p \in P$, we can derive p from the given distance vector $D_R(p)$. We will split it into parts $D_{R_i}(p)$, projected onto the respective R_i components. Note that since $R_i \subset K(P_i)$, all points in R_i can see all points in P_i. Thus if an entry of $D_{R_i}(p)$ is -1, we can conclude that $p \notin P_i$. If this is the case, we will say that P_i is an impossible location for p, else we consider it plausible. Observe that there must be at least one plausible P_i, since we have that p is in some P_i, we just do not know which yet.

Suppose there is only one plausible P_i. Then, the same process used to trilaterate P_i with R_i can be reused.

Suppose instead that there are at least two plausible P_i, P_j. Then all guards in $R_i \cup R_j$ can see p and yield distance values. Since $R_i \cup R_j$ contains three non-collinear guards, we can solve for the location of p. □

We will show how to find such a partition for a general polygon P which satisfies Lemma 1. In particular, we will partition using only diagonals of P. For each piece and corresponding guard placement (P_i, R_i), we will have either R_i being three non-collinear guards, or two guards on an edge of P_i, which is a diagonal or edge of P. If we make sure not to reuse the same diagonal for different placements, this will limit our placement to distinct diagonals. Our first argument will assume that P is in general position, meaning the above placement satisfies Lemma 1. Our second argument will address the case when diagonals of P may be collinear, and show that, as long as we are still using distinct diagonals for placement, an application of Observation 2 will let us locate all points in P.

Observation 3 *If P is in general position, and a, b and c, d are on distinct diagonals, then a, b, c, d are not collinear.*

3 Upper bound for general position polygons

Before we state our main theorem, we first show a trivial upper bound and lower bound. Suppose we put a distance measuring guard on each vertex of a polygon with N vertices. If we consider its triangulation, then we can see that any point in P can always view three non-collinear vertices of P. Hence it can see three non-collinear guards, and can be trilaterated. Thus N guards always suffices for any polygon.

To show a lower bound, we can reuse the comb polygon lower bound for the art gallery problem. No point in the comb can view into two different comb spikes. Thus we need at least two guards per spike to trilaterate all of the points in its interior. Thus $2N/3$ is a lower bound for minimum trilateration.

We now present the main theorem of this paper, which is giving an upper bound better than N guards:

Theorem 2 *Any simple polygon P with N vertices can be trilaterated by a guard set of size no more than $8N/9$.*

To prove Theorem 2, we give both a method of constructing a guard set, and a method for inverting distance vectors back to points. Our method has several steps. First, we use a generic fan partition to divide P into $N/3$ pieces, each piece being star shaped, and some having several prospective edges (which are either diagonals or edges of P) on which we could place guards. Second, we use a bipartite graph connecting pieces of the partition to possible diagonals / edges for placement of guards, preventing separate pieces from using the same diagonal for placement. After finding a maximum matching in this graph, we return, along with each piece of the partition, the set of guards which trilaterate that piece. The pseudocode for this algorithm is presented later on as Algorithm 1. For inverting these distances, we still use the procedure presented in Lemma 1, the pseudocode for which is presented as Algorithm 2.

The partition that we use is Chvátal's fan partition from the original proof of the art gallery theorem [5]. Although it is classic, it has largely been overshadowed by the Fisk proof using coloration [7]. Thus we will restate it here:

Definition 2 *A fan is a polygon P with at least one vertex u, such that for all other vertexes v not adjacent to u, $\overline{uv}$ is a proper diagonal of P. We call u the center of the fan.*

Theorem 3 *Every N-triangulation can be partitioned into m fans where $m \leq \lfloor n/3 \rfloor$. Furthermore, this can be done in $O(N \log N)$ time.*

This theorem is useful as it allows us to use any triangulation method we wish in order to find a fan partition. Thus our algorithm is agnostic to the method used to find the triangulation. Our run time is essentially dominated by the $O(N \log N)$ time needed to convert the triangulation into a fan partition. Thus, we can use the standard $O(N \log N)$ triangulation method. See Figure 1 for an example of a polygon partitioned into fans.

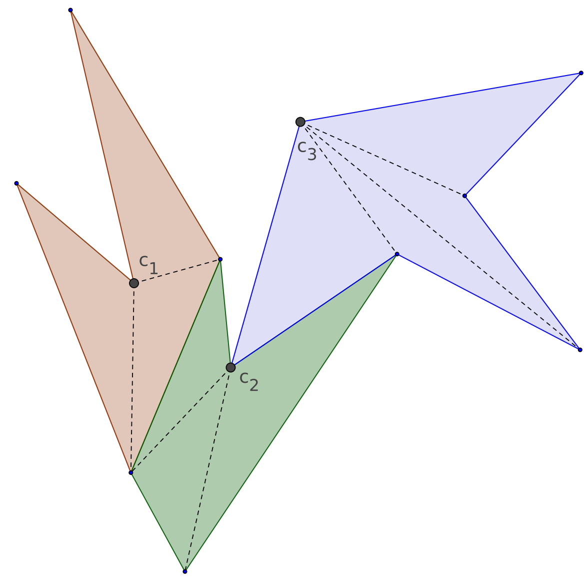

Figure 1: A polygon partitioned into three fans, with centers labeled c_1, c_2, c_3.

3.1 Trilaterating fans

We show several important structural lemmas. Mainly, we show that every fan with 4 or more triangles can be trilaterated by 3 guards, while every fan with fewer than 4 triangles can be trilaterated by 2 guards on an edge.

We refer to a fan with k triangles in its triangulation as a k-fan.

Definition 3 *For an edge e of a polygon P, we say that e is a prospective edge if there are two points $a, b \in e$ in the kernel of P. As such, any polygon with a prospective edge can be trilaterated with 2 guards.*

Lemma 4 *Every fan of k edges can be trilaterated with 3 guards, which can be found in $O(k)$ time.*

Lemma 5 *Every 1-fan has 3 prospective edges, every 2-fan has at least 2 prospective edges, and every 3-fan has at least 1 prospective edge. Further, these edges can be found in $O(1)$ time.*

We defer the proof of these structural lemmas to the appendix.

3.2 Finding a diagonal disjoint placement

We have shown that every fan can be trilaterated, either with three guards in its kernel, or two guards on

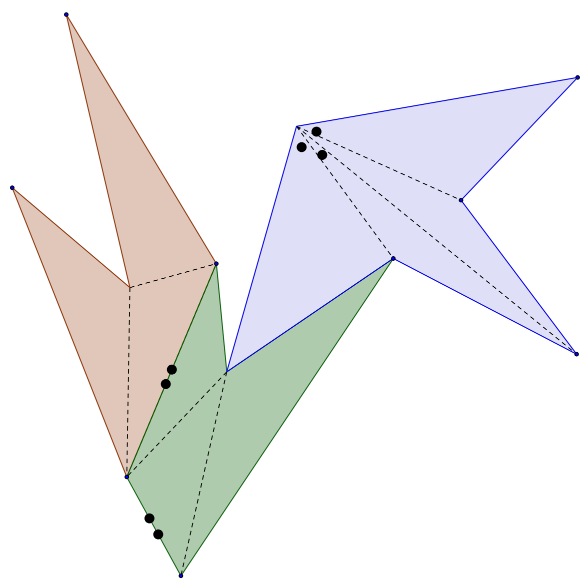

Figure 2: The fan partition from Figure 1 with guards covering each fan.

its boundary. Such a boundary placement corresponds to a placement on a diagonal or edge of P. In order to maintain the requirement of Lemma 1, we must find a placement such that no two fans put their guards on the same diagonal.

To do this, our algorithm uses the following steps:

- Partition P into a set F of no more than $N/3$ fans.
- For each fan f, associate with it all diagonals or edges which intersect $K(f)$ in at least two points. We do not bother attempting this step if f has 4 or more triangles.
- Given each fans' prospective edges, find a max matching between fans and edges.
- For each fan, report an appropriate pair of guards if it was matched to an edge, or triple of guards inside its kernel otherwise.

To do this, we create a bipartite graph $G = ((F \cup D), E)$, where F is the set of fans, and D is the set of diagonals and edges of the partition. We add the edge (f, d) if d is a viable boundary edge for fan f. We note a few properties of the graph. First, all 3-fans have degree at least 1, 2-fans at least 2, 1-fans exactly 3, and 4+-fans exactly 0. Second, if $d \in D$, then $deg(d) \leq 2$. Lastly, Because our graph is defined by the partition, it is a tree. Note that in finding a matching on this graph and reporting approprite guards, we do not guarantee that a fan which could have been trilaterated with 2 guards will still only use 2 in our placement. However, this is necessary in order to satisfy Lemma 1. See Figure 2 for an example of a valid placement of guards in a fan partition which could result from this algorithm.

Consider the number of fans using i triangles, for $i = 1, 2, 3, 4, \ldots\ldots$ Call these sets F_i. As previously shown in Lemma 5, all fans in F_1 have at least 3 prospective edges, F_2 at least 2, and F_3 at least 1. Hence these nodes in the bipartite graph will have degrees at least 3, 2, and 1 respectively. Fans with 4 or more triangles we will simply cover with 3 guards and not pair with any diagonals. Any diagonal can border at most 2 distinct fans, so every diagonal node in the graph will have degree no more than 2.

Suppose that we had k fans, so that $|F| = k$, and we found a matching of size j. Then, for j of the fans, we could use 2 guards, and for the remaining $k - j$ fans we would use 3 guards. This makes our total guard usage $2j + 3(k - j) = 3k - j$. Since Theorem 3 implies $k \leq \lfloor \frac{N}{3} \rfloor$, we are using no more than $N - j$ guards. We thus minimize our guard count by maximizing j.

We now present the main lemma regarding our graph construction, which we present as a generalized result on bipartite graphs with certain degree constraints:

Lemma 6 *Let $G = (A \cup B, E)$ be a bipartite graph, satisfying that for all nodes $v \in B$, $deg(v) \leq 2$. Let c_0, c_1, c_{2+} be the sets of nodes in A with degrees 0, 1, and ≥ 2, respectively, as well as the sizes of these sets. Then a matching of size $\frac{c_1}{2} + c_{2+}$ is always possible.*

Proof. We prove the above by an application of the deficit version of Hall's Theorem [10]. Let $S \subset A$. Then the defect of S is defined as $df(S) = |S| - |N(S)|$, the size of S minus the number of unique neighbors of S in B. Then the maximum matching M satisfies $|M| = \min_{S \subset A}\{|A| - df(S)\}$.

We will upper bound $df(S)$, which gives us a lower bound on $|M|$. Consider an arbitrary such S. Let m be the number of edges leaving S, and let $d(k)$ be the number of nodes of degree k in S. Then $|N(S)| \geq m/2$, as each node in B has degree at most 2. Also note that $m = \sum kd(k)$, and $|A| = c_0 + c_1 + c_{2+}$.

$$\begin{aligned}
|S| - |N(S)| &\leq |S| - m/2 \\
&= \sum d(k) - 1/2 \sum kd(k) \\
&= \sum (1 - k/2)d(k) \\
&\leq d(0) + d(1)/2 \\
&\leq c_0 + c_1/2
\end{aligned}$$

Hence $df(S) \leq c_0 + c_1/2$, and $|M| \geq A - (c_0 + c_1/2) = c_1/2 + c_{2+}$, as desired. □

We can now prove the main theorem of the paper, restated below:

Theorem 2 *Any simple polygon P with N vertices can be trilaterated by a guard set of size no more than $8N/9$.*

Proof. We can partition P into $k \leq N/3$ fans, associate with each fan its plausible diagonal placements, and find a matching on the graph G as above. For any set $R_i \cup R_j$, either R_i contains three non collinear guards, or R_i and R_j are on distinct, non-collinear diagonals. Thus the placement satisfies Lemma 1 and trilaterates P.

The matching of G will have size at least $c_1/2 + c_{2+} = F_3/2 + F_2 + F_1$. Thus, the number of guards we will use is no more than $3k - F_3/2 - F_2 - F_1$. Thus we can bound the number of guards returned by analyzing the linear program:

$$\begin{aligned} \max \quad & 3k - F_3/2 - F_2 - F_1 \\ \text{s.t.} \quad & F_1 + F_2 + ... = k \\ & k \leq \lfloor N/3 \rfloor \\ & F_1 + 2F_2 + 3F_3 + ... = N - 2 \\ & F_i \geq 0 \end{aligned}$$

It can be shown via standard dual arguments that for all N the objective function is bounded above by $8N/9$. See the appendix for a derivation of this bound. Thus we can always achieve less than $8N/9$ guards. $\square$

We now present the algorithm pseudocode for placement and for location. Although the lemma for locating points provides an algorithm, we will explicitly give it as pseudocode, in order to generalize our results to arbitrary simple polygons in the next section.

Lemma 7 *Algorithm 1 runs in $O(N \log N)$ time.*

We defer the proof to the appendix, as it is a routine examination of the algorithm step by step.

4 Extending the upper bound to polygons that are not in general position

Consider now a polygon which may not be in general position. The argument that guards on distinct diagonals can trilaterate points is no longer valid, as distinct diagonals may be collinear. We show how to augment the the location step with an additional method, in order to still locate points, even if the only guards that can see them are all collinear. To do this, we use the following lemma:

Algorithm 1: guard Locations Algorithm

```
Input  : Polygon P
Output: Partition of P with corresponding guard
        placements
T = Any triangulation of P
F = FanPartition(T)
D = Edges of F
f = Faces of F
G = (f ∪ D, {})
for f_i ∈ f do
    d_i ← viable diagonal edges for f_i
    Add edge (f_i, d) to G for all d ∈ d_i
end
M = MaxMatching(G)
for (f_i, d_i) ∈ M do
    r_1, r_2 ← two points ∈ d_i ∩ K(f_i)
    Yield (f_i, {r_1, r_2})
end
for f_i ∉ M do
    r_1, r_2, r_3 ← three non-collinear points ∈ K(f_i)
    Yield (f_i, {r_1, r_2, r_3})
end
```

Algorithm 2: Distance Vector Reversing

```
Input  : Polygon P, Partition {P_i, R_i}, Distance
        Vector D
Output: Unique point p that generates D
for {P_i, R_i} in partition do
    D_i ← D projected onto R_i
    if D_i has a −1 entry then
        Disregard {P_i, R_i}
    end
end
if At least two (P_i, R_i), (P_j, R_j) remain then
    locatePoint(R_i ∪ R_j, D_i ∪ D_j)
end
else
    locateWithinP(P_i, R_i)
end
```

Lemma 8 *Let $r_1 \neq r_2$ be points in P such that the line segment $e = \overline{r_1 r_2}$ intersects $\delta(P)$, the boundary of P. Then, the intersection of visibility regions $V(r_1) \cap V(r_2)$ is entirely in one closed half plane induced by $\overline{r_1 r_2}$.*

We defer the proof of the above to the appendix. The application of this lemma is that, if r_1 and r_2 are on distinct diagonals, at least one point directly between them is on the border of P. Thus if $p \in V(r_1) \cap V(r_2)$, we can narrow down its location to a specific half plane defined by r_1, r_2.

Suppose we had a method which took queries of pairs of collinear diagonals d_l, d_r, and returned which of their sides cannot have common visibility to both. Then, updating our location method involves only updating the method **locatePoint**. First check if the input guards are collinear. If they are not, solve the system as before. If they are collinear, then since they are on distinct diagonals, we determine on which side of these diagonals p cannot be in. Once we know this side, we can solve the system for two potential points, and return the point that is on the correct side.

Our proposed method will require us to maintain the triangulation structure from the placement part of the algorithm, and perform path finding in the dual graph. Let d_1, d_2 be collinear diagonals in a triangulation T, and consider how the graph dual of T is partitioned by the edge corresponding to d_1. Then the piece of the partition which has d_2 contains the piece of the polygon which can have common visibility to d_1 and d_2. We can make a similar claim on d_2. To determine which side of d_1 this piece is on, use the triangulation structure to determine the triangle which uses d_1 and is in the same piece of the partition as d_2. Use a clockwise test to return whether the third point of this triangle is to the right or the left of d_1. The piece of P with common visibility must also be to the right or to the left of d_1. See Algorithm 3 for the pseudocode for this method.

Algorithm 3: General Position locatePoint

```
Input  : Polygon Triangulation T, Guard
         positions R, Distance Vector d
Output: Unique point p that generates d
P = All solutions to system ||r_i − d_i|| = p
if |P| = 1 then
  | Return the unique p ∈ P
end
else
  | P = {p_1, p_2}
  | Take r_i, r_j ∈ R on distinct diagonals D_i, D_j
  | Let t be the first triangle in the unique path
  | between D_i and D_j in T
  | Let x be the vertex of T not on D_i
  | if ccw(r_i, r_j, x) == ccw(r_i, r_j, p_1) then
  |   | return p_1
  | end
  | else
  |   | return p_2
  | end
end
```

5 Conclusion

We introduced the minimum trilateration problem, and showed that it has an upper bound of $8N/9$ guards. We gave an $O(N \log N)$ algorithm for achieving this bound, as well as an algorithm for using the given placement to invert distance vectors to locate points in the polygon.

Having introduced the trilateration problem and derived an upper bound similar to that for the art gallery problem, there are several questions left unanswered. The ones we are most interested in are finding a tight upper bound, giving an algorithm to verify proposed guard sets, and showing improved bounds for the usual variants of art gallery, such as orthogonal polygons, guards which can move along edges, or guards which can see through k walls.

References

[1] D. Avis and G. T. Toussaint. An optimal algorithm for determining the visibility of a polygon from an edge. *IEEE Trans. Comput.*, 30(12):910–914, Dec. 1981.

[2] M. d. Berg, O. Cheong, M. v. Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.

[3] Daniel Busto, William S. Evans, and David G. Kirkpatrick. On k-guarding polygons. In *Proceedings of the 25th Canadian Conference on Computational Geometry, CCCG 2013, Waterloo, Ontario, Canada, August 8-10, 2013*, 2013.

[4] J. Cáceres, M. C. Hernando, M. Mora, I. M. Pelayo, M. L. Puertas, C. Seara, and D. R. Wood. On the metric dimension of cartesian products of graphs. *SIAM J. Discrete Math.*, 21(2):423–441, 2007.

[5] V. Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory Series B*, 18:39–41, 1975.

[6] K. Curran, E. Furey, T. Lunney, J. Santos, D. Woods, and A. McCaughey. An evaluation of indoor location determination technologies. *J. Location Based Services*, 5(2):61–78, 2011.

[7] S. Fisk. A short proof of Chvátal's watchman theorem. *J. Comb. Theory, Ser. B*, 24(3):374, 1978.
W. H. Freeman & Co., New York, NY, USA, 1979.

[8] W. Goddard and O. R. Oellermann. Distance in graphs. In M. Dehmer, editor, *Structural Analysis of Complex Networks*, pages 49–72. Birkhuser Boston, 2011.

[9] D. T. Lee and F. P. Preparata. An optimal algorithm for finding the kernel of a polygon. *J. ACM*, 26(3):415–421, July 1979.

[10] Oystein Ore. Graphs and matching theorems. *Duke Math. J.*, 22(4):625–639, 12 1955.

[11] J. O'Rourke. *Art Gallery Theorems and Algorithms*. The International Series of Monographs on Computer Science. Oxford University Press, New York, NY, 1987.

[12] J. Urrutia, U. Nacional, and A. Mxico. Art gallery and illumination problems. In *In Handbook on Computational Geometry, Elsevier Science Publishers, J.R. Sack and*, page 1026, 2000.

Appendix

We provide the omitted proofs for several lemmas.

Lemma 4 *Every fan of k edges can be trilaterated with 3 guards, which can be found in $O(k)$ time.*

Proof. First, note that if a vertex can view all other vertices, it can view all other points in the polygon [1]. This asserts that all fans have a non-null kernel containing the center vertex v. We will now examine the kernel as the intersection of k half-planes.

Each edge of the polygon defines a half plane, the intersection of which is the kernel of the polygon. The two edges adjacent to v will map to half planes which intersect v. Since these half planes cannot be parallel or anti-parallel (defining opposite half-spaces of the plane), their intersection is a region which v is on the border of. Observe that every other half plane must contain v, since v is in the kernel, and that every other half plane cannot intersect v on its boundary. To see why, consider the half plane defined by the edge $\overline{ab}$. We have that both $\overline{av}$ and $\overline{bv}$ are non-intersecting valid diagonals of P. Since they are not intersecting, v is not collinear with $\overline{ab}$. Hence the edge of that half plane is at least some distance ϵ away from v.

Hence if we take the intersection of all these half planes, the result is a region around v with an infinite number of points. Taking any 3 of them as the guard set will suffice.

To find these guards, the linear time kernel algorithm can be used to determine the kernel [9], from which we return several random non-collinear points. If we wished for a slightly simpler algorithm, we do not need the explicit kernel, but just a subset of it. We calculate the distance of v from each half plane, and take the min of these as r. Then any point which is in the intersection of the $\overline{av}$ and $\overline{bv}$ half planes and within distance r of v will be in the kernel. It suffices to pick one and move it $\pm\epsilon$ to get three non-collinear points.

□

Lemma 5 *Every 1-fan has 3 prospective edges, every 2-fan has at least 2 prospective edges, and every 3-fan has at least 1 prospective edge. Further, these edges can be found in $O(1)$ time.*

Proof. The lemma is clearly true of a 1-fan since it is a triangle which is convex. Thus all edges are prospective edges.

A 2-fan is a quadrilateral. Let v be the center of the fan, a and b the vertices adjacent to v, and c the last vertex. Note that the angles at a and b must be convex. Consider the visibility regions $V(a)$ and $V(b)$. First note they must intersect an edge not adjacent to a and b respectively. Hence $V(a)$ contains $\overline{ac}$ and intersects part of $\overline{bc}$, while $V(b)$ contains $\overline{bc}$ and intersects part of $\overline{ac}$. Thus some sections of $\overline{bc}$ and $\overline{ac}$ are in the kernel, making them prospective edges.

A 3-fan is a pentagon. Consider the vertices of the fan labeled in CW order a, v, b, c, d, so that a and b are adjacent to v. Note that the angles at a and b must be convex. If the angle at v is convex, then following Lemma 4, the kernel intersects both $\overline{va}$ and $\overline{vb}$. Else, we consider the case where v is a reflex angle. Since a pentagon can have at most two reflex angles, at least one of c or d is a convex angle. WLOG assume c is convex.

Suppose that d is a reflex angle. Then, we have that we can extend both $\overline{ad}$ and $\overline{av}$ until they hit the boundary of the fan. They must both end at $\overline{bc}$, creating a subsegment which can see every vertex. Thus $\overline{bc}$ is a prospective edge.

Suppose instead that d is a convex angle. Then instead consider extending $\overline{av}$ and $\overline{bv}$ until they hit the boundary of the fan. If $\overline{av}$ hits $\overline{bc}$, then because d is convex, c can see a. Thus c could also be the vertex center, and we can reduce to the case where it is a fan with convex angle at the center. A similar argument applies to if $\overline{bv}$ hits $\overline{ad}$. Thus, we must have that both $\overline{av}$ and $\overline{bv}$ extend to meet $\overline{cd}$. Then similar to the previous case, they create a subsegment which can see every vertex. Thus $\overline{cd}$ is a prospective edge. Thus in all cases, a pentagon has a prospective edge.

To find these edges, we can explicitly compute the kernel, and determine which edge of the polygon it intersects with in a continuous region. Since our fan size is no more than 5 edges, this takes $O(1)$ time.

□

Lemma 1 *Algorithm 1 runs in $O(N \log N)$ time.*

Proof. Triangulating the polygon can be done in $O(N \log N)$ time. 3-coloring this triangulation and determining the resulting fan partition can be done in $O(N \log N)$ time, by computing the DCEL representation of the partition [2], and considering the bounded faces to be the fans.

For each fan of k edges, either determining the prospective edges or finding three points in the kernel takes $O(k)$ time. Thus summing over all fans in the partition, the total time from these operations is $O(N)$.

To find the maximum matching in our bipartite graph, it is known that a maximum matching on trees can be found in $O(N)$ time by using dynamic programming. Since our graph was induced by the edges of the polygon and diagonals of the fan partition, it is a tree.

Thus the dominating factor in the run time is using the diagonals of the triangulation to retrieve the DCEL representation of the partition, which takes $O(N \log N)$ time.

□

Theorem 2 *Any simple polygon P with N vertices can be trilaterated by a guard set of size no more than $8N/9$.*

Proof. We derive an upper bound for the linear program, which we restate here:

$$\begin{aligned} \max \quad & 3k - F_3/2 - F_2 - F_1 \\ \text{s.t.} \quad & F_1 + F_2 + ... = k \\ & k \le \lfloor N/3 \rfloor \\ F_1 + 2F_2 + 3F_3 + ... &= N - 2 \\ & F_i \ge 0 \end{aligned}$$

First note that our objective function is equivalent to $2F_1 + 2F_2 + 2.5F_3 + 3F_4 + 3F_5 + ...$ We show that a linear combination of the constraints bounds our objective function.

Take 5/3 of the first inequality, and add it to 1/3 of the second equality, yielding:

$$\begin{array}{r} (5/3)(F_1 + F_2 + F_3 + ...) \le (5/3)\lfloor N/3 \rfloor \le 5N/9 \\ (1/3)(F_1 + 2F_2 + 3F_3 + ...) = (N-2)/3 \\ \hline 2F_1 + (7/3)F_2 + (8/3)F_3 + 3F_4 + ... \le (8N-6)9 \end{array}$$

Where the LHS of the final inequality is greater than our objective function. Thus our objective function is bounded above by $8N/9$. □

Lemma 2 *Let $r_1 \neq r_2$ be points in P such that the line segment $e = \overline{r_1 r_2}$ intersects $\delta(P)$, the boundary of P. Then, the intersection of visibility regions $V(r_1) \cap V(r_2)$ is entirely in one closed half plane induced by $\overline{r_1 r_2}$.*

Proof. We will consider a coordinate frame of axis where r_1 and r_2 are both on the x-axis and have opposite signs for their x coordinate. First, we consider the case where either r_1 or r_2 is on an edge e of P. In our coordinate frame, e is coincident with the x-axis. In this case, it should be clear that, depending on which side of e P is on, WLOG r_1 can only see above or below the x axis, but not both. Hence the intersection of visibility regions must be above or below the x-axis, as desired.

Let $X \neq r_1, r_2$ be a point in $\delta(P)$ which is on e. Take a point with arbitrarily small x coordinate $Z = (-\infty, 0)$, and consider any closed curve path from Z to X which does not intersect P before touching X. Let Y be the first point on e which P meets, which may end up being Z. Then, we now consider the close curve path P', which is P from Z to Y. Consider the side of e that P'approaches Y from. Since there is a path from Z to Y that lies completely outside the polygon, $V(r_1)$ and $V(r_2)$ cannot intersect on that side of e, as this would mean that they also intersect P', which lays entirely outside the polygon.

□

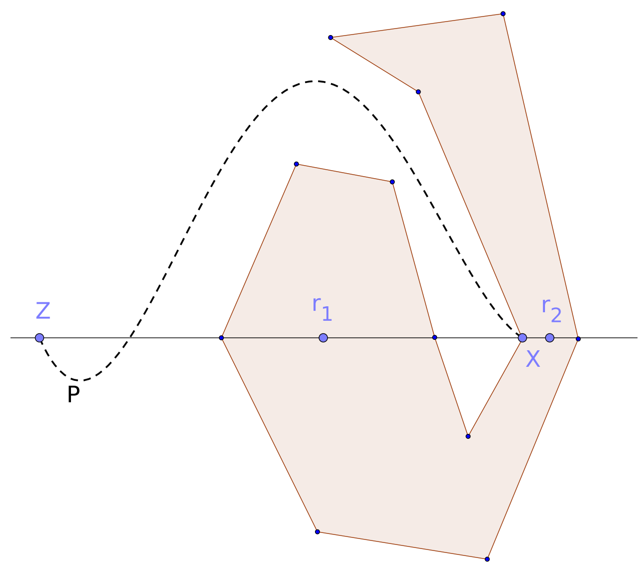

Figure 3: An illustration of Lemma 2. The existence of the path P guarantees us that $V(r_1)$ and $V(r_2)$ cannot intersect above the x-axis.