

A Gradient Search Algorithm for the Maximal Visible Area Polygon Problem

Helman I. Stern¹, Moshe Zofi^{1,2}

¹Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beersheva, Israel

²Department of Industrial Management, Sapir College, Sderot, Israel
Email: helman@bgu.ac.il, zofi@bgu.ac.il

Received 28 March 2015; accepted 8 May 2015; published 12 May 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper provides a gradient search algorithm for finding the maximal visible area polygon (VAP) viewed by an interior point in a simple polygon P . The algorithm is based on a natural partition of P into convex sets, such that each element of the partition is associated with a unique analytical form of the area function. We call this partition a back diagonal partition of P . Our maximal VAP algorithm converges in a finite number of steps, and is polynomial with a complexity of $O(r^2 n^2)$, for a simple polygon P with n vertices, and r reflex vertices. We use the maximal VAP algorithm as a basis for a greedy heuristic for the well known guardhouse problem with a computation complexity of $O(r^3 n^2)$.

Keywords

Maximal Visible Polygon, Gradient Search, Continuous Optimization, Guardhouse Problem

1. Introduction

A visual polygon is a polygon in which any pair of points is mutually visible. A visible polygon from a point x in a simple polygon P with n vertices can be found with a computational complexity of $O(n)$ by the algorithm of El Gindy and Avis [1]. For each visible polygon, one may associate a real number representing its area. The area function over all points $x \in P$ is not well behaved, being in general neither convex nor concave, including points that may be not differentiable and even discontinuous. Some properties of polygonal area functions may be found in [2].

The first problem we intend to solve is finding an interior point x that maximizes the area of the visual po-

lygon. We denote this problem as the maximal visible area polygon (VAP) problem. This problem was studied by Cheong *et al.* [3], which suggested sampling a finite number of points inside the polygon found by its triangulation.

The algorithm we suggest for this problem is based on a gradient search which converges with a finite number of steps. In order to determine the size of each step taken in the gradient direction, we use a specialized partition of the polygon P into convex sets referred to as a back diagonal partition (BDP). This decomposition arises naturally through the introduction of “back diagonals” in such a manner that the area functional expression is invariant within the domain of each element of the partition. Gradient information along with jump points is combined in an algorithm which traces a path from some initial point in the polygon to a local maximal point. The developed algorithm has a computational complexity of $O(r^2n^3)$ for a polygon P with n vertices and r reflex vertices. Such a path may be used to guide a mobile search robot in an enclosed 2D area.

A related problem of placing the minimum number of stationary guards to visually cover a room represented as a 2D polygon was first presented by Klee in 1973 [4] as the art gallery or guardhouse problem. The objective was to find the smallest set of points in a polygon, such that every point inside the polygon would be visible to at least one point in the set. Two points are visible to each other, if the line of sight connecting them lies inside the polygon. Several heuristics and complexity calculations are provided in [5] and [6] for different extensions of the guardhouse problems in which the observers are restricted to the vertices of the polygon, and the edges of the polygon, or not restricted at all. In this paper, we offer a greedy algorithm, which repeatedly uses the gradient search method for finding the maximal VAP, and for solving the guardhouse problem. This greedy heuristic terminates after a finite number of steps and is polynomial with time complexity of $O(r^3n^2)$. Although the original problem was described in terms of the positioning of guards, it had application to the positioning of surveillance cameras, motion sensors or other detection devices in enclosed environments. We assume that omni vision cameras (or several back to back cameras) are placed at the selected positions. This problem has recently become a popular security problem. For example, currently England has 5 million CCTV security cameras and is adding them at the rate of 100 per day in both inside and outside environments.

In Section 2, we define the maximal VAP problem and introduce useful notation. In Section 3, we introduce the concept of a hidden region, its area, and a back diagonal partition. These set the stage for the introduction of the maximal visible area polygon (VAP) gradient search algorithm described in Section 4. In Section 5, we use the gradient maximal VAP algorithm to form a basis for a greedy algorithm to solve an application in the form of the guardhouse problem, and illustrate it with a small example. Section 6 provides conclusions and future work.

2. Problem Definition and Notation

Before embarking on a formal definition of the problem we present a number of definitions and associated notation. A polygon P is a closed plane figure bounded by at least three straight connected line segments. A polygon P containing n line segments can be described using the vertices along its boundary $P = \{v_1, v_2, \dots, v_n\}$. Let e_i denote a *directed edge* between adjacent vertices of P where $e_i = (v_i, v_{i+1})$, $i = 1, \dots, n$ and $n+1 = 1$. A *simple polygon* is a polygon which does not contain any holes and whose edges do not intersect. Two points u and v in P are mutually visible if for all points w such that $w = \lambda u + (1-\lambda)v$, $\lambda \in [0, 1]$, lie within P . A vertex of P is said to be non-convex or *reflexive* if the angle between the two segments that share it as endpoints is greater than 180° when viewed from the interior of P . *Reflex vertices* can create hidden areas when viewed from some points inside the polygon. **Figure 1(a)** shows a simple polygon where vertex v_6 is a reflex vertex, inducing a hidden area with respect to an observation point.

We define $V(p, P)$ as a *visibility polygon*, such that each point in V is visible by an observation point p . The area of V is denoted as $A(V(p, P))$. When understood these will be written simply as $V(p)$ and $A(p)$. Given a polygon P of size n , and a point p inside P , the visible polygon $V(p, P)$ and its area $A(V(p, P))$ can be determined with a time complexity of $O(n)$ [1].

The Maximal VAP Problem

Define a maximal visible area polygon (VAP) problem as finding a point p^* for which $A(V(p, P))$ is maximal, *i.e.*

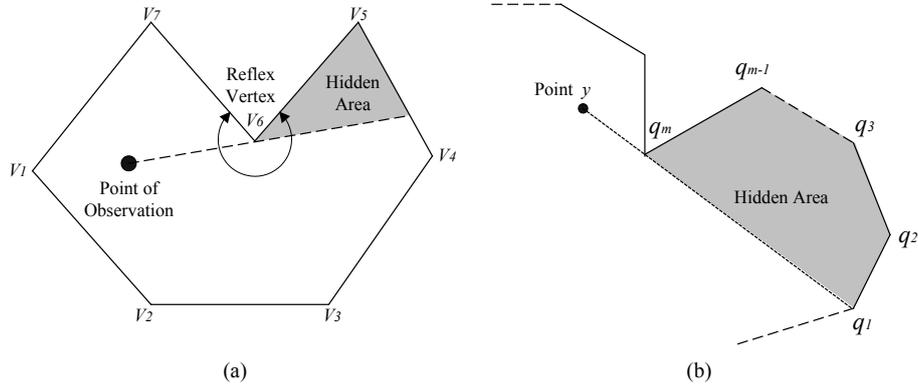


Figure 1. Simple polygons with hidden areas. (a) A simple polygon containing a reflex vertex; (b) A region created by a chain of vertices which are not visible from an observation point.

$$\text{Max}_{p \in P} A(V(p, P)) = A(V(p^*, P))$$

The expression for the signed area of a polygon P , with vertex coordinates $\{(x_i, y_i), i = 1, \dots, n\}$ reduced modulo n $y_0 = y_n$, $y_{n+1} = y_1$, is

$$A(P) = \frac{1}{2} \sum_{i=1}^n x_i (y_{i+1} - y_{i-1})$$

Note, that $A(P)$ represents the signed area of the polygon P , i.e. $A(P) > 0$, if the vertices of P are indexed in counterclockwise order. The visible area function over all $x \in P$ is not well behaved, being in general neither convex nor concave. Moreover, it may contain points that are not only non differentiable but discontinuous. For the example in **Figure 2(c)**, the top flat square has a non differentiable boundary and discontinuous drops at its corners.

3. Local Maximal Visible Area Polygon (VAP) Gradient Search Algorithm

We propose a gradient search method which starts with a given observation point, and moves it in the direction of the maximal gradient until a local maximum is achieved. At each step of the algorithm, a different visible area is calculated (i.e. the polygon minus the hidden regions) as a function of the current point. The result of a step in the maximal area gradient direction will give the (x, y) coordinates of the new observation point. Since there are several local gradients (one for each hidden region), the final direction is determined by the resultant vector. For example, if polygon P has two hidden regions with respect to a given point p , then two gradients are calculated (one for each hidden area). If the gradients for hidden regions I and II are (x_1, y_1) and (x_2, y_2) , respectively, then the final gradient (∇_x, ∇_y) will be $(x_1 + x_2, y_1 + y_2)$. Below we present several definitions, which will form the basis of a gradient search method to be described subsequently.

3.1. Basic Definitions and Notation

Given a simple polygon $P = \{v_1, v_2, \dots, v_n\}$ with vertices ordered in counterclockwise order, for any three consecutive vertices $v_1 = (x_1, y_1)$, $v_2 = (x_2, y_2)$, $v_3 = (x_3, y_3)$ define a turn: T as:

$T = (y_3 - y_2) \cdot (x_2 - x_1) - (x_3 - x_2) \cdot (y_2 - y_1)$. We say that $v_1 v_2 v_3$ is a left turn if T is positive, a right turn if T is negative, and not a turn otherwise.

Proposition 1 Let a chain Q be a sequence of connected, nonintersecting line segments. Given two points y and z , such that the line segment yz does not intersect any line segment of Q , then y and z are said to be mutually visible with respect to chain Q .

Proposition 2 Let $Q = \{q_1, q_2, \dots, q_m\}$ be a simple chain such that q_1 and q_m are visible with respect to Q . If a point y is exterior to the region enclosed by (Q, q_1) and collinear with q_1 and q_m , then the region

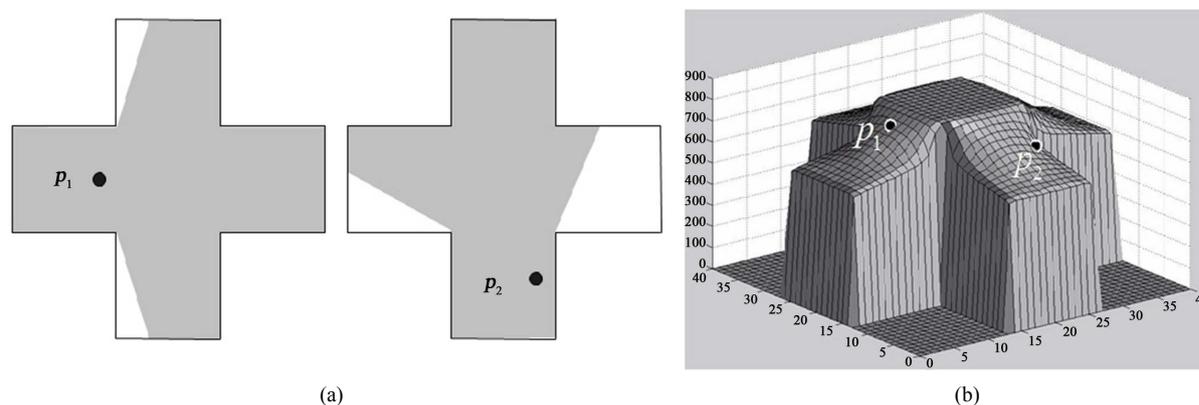


Figure 2. A cross shaped polygon. (a) The visible polygon from point p_1 (area = 93.6%); (b) The visible polygon from point p_2 (area = 82.9%); (c) The area function.

enclosed by (Q, q_1) is not visible from y . The proof appears in [1].

Figure 1(b) shows an example of a point y exterior to a chain $Q = \{q_1, q_2, \dots, q_{m-1}, q_m\}$. The point y is collinear with q_1 and q_5 , and therefore, cannot see the region enclosed by (Q, q_1) . A line segment $q_1 q_m$ is said to be a window of y with respect to the region enclosed by (Q, q_1) . This region is said to be a hidden region with respect to y . Note, that if q_m is visible from y then so is q_1 . We denote the window, $W = (q_1, q_m)$ of y , as a left hand window (LHW) if $yq_m q_{m-1}$ is a left turn, and as a right hand window (RHW) if $yq_m q_{m-1}$ is a right turn. Looking back at **Figure 1**, we can see that $yq_m q_{m-1}$ is a left turn. Let p be a viewpoint inside P and s a reflex vertex on the boundary of P . We classify reflex vertices as active or inactive according to the possibility of each creating an area occluded from p . In order to determine whether a reflex vertex s is active or inactive with respect to a given viewpoint p , we extend the adjacent edges of s into the interior of P until they hit the boundary to obtain back diagonals χ_1 and χ_2 . If p and s are mutually visible and p lies outside the angle χ_1, s, χ_2 , then s is said to be an active reflex point, otherwise s is inactive. Given a viewpoint p and an active reflex point s , define a ray Δps from p in the direction of s and passing through s . Let $I = \{p_1, p_2, \dots, p_q\}$ be the set of intersection points of the ray with the boundary of P , ordered according to the distance from p . Denote the intersection point closest to p as a reflection point p' . **Figure 3(a)** shows the reflection point p' (the first intersection point) with respect to a viewpoint p and an active reflex vertex s .

3.2 Back Diagonal Partition (BDP)

Consider a pair of vertices u and v on the boundary of P , such that u is a reflex vertex and v is visible from u . Define a diagonal uv as a line segment directed from vertex u to vertex v . A *back diagonal* uv' with respect to uv is a line segment joining the vertex u and the point v' , where v' is the nearest point on the edge of P visible in the reverse direction of uv . The point v' is said to be a reflection point on the boundary of P visible from v along a ray directed from v through u . **Figure 4** shows a back diagonal uv' created by a reflex vertex u and a vertex v visible from u .

Figure 5 shows an example of all back diagonals (dotted lines) induced by the reflex vertices. The introduction of all back diagonals induces a *back diagonal partition* (BDP) of P . Such a partition is a planar graph defining a set of internal regions. Note that five back diagonals are induced by vertex v_{10} —one for each of the vertices in P visible from v_{10} .

Proposition 3 A BDP of P partitions P into a collection of regions, each a convex sub polygon of P .

Proposition 4 A back diagonal is an edge in a BDP such that viewpoints in the neighborhood on either side of it have different visible and area polygonal functional expressions. Thus, each region in a BDP has a unique visible and area polygonal functional expression.

The above propositions will be useful in construction an algorithm for the maximal VAP problem.

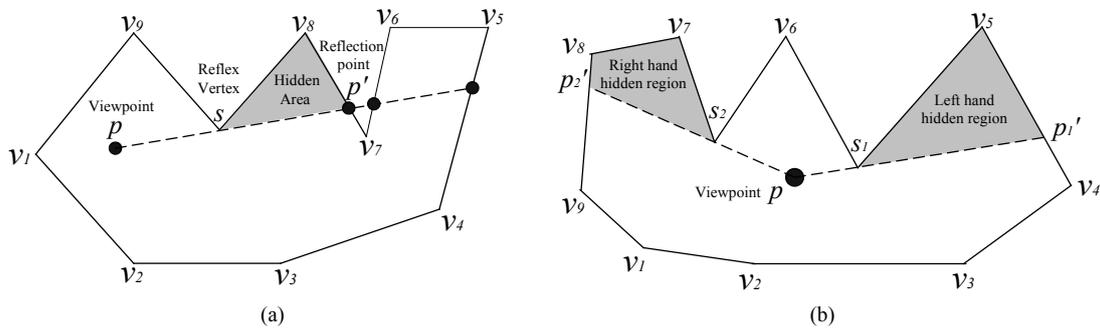


Figure 3. (a) A reflection point p' , with respect to a given viewpoint p and an active reflex vertex s ; (b) Left and right hand hidden regions with respect to a given viewpoint p .

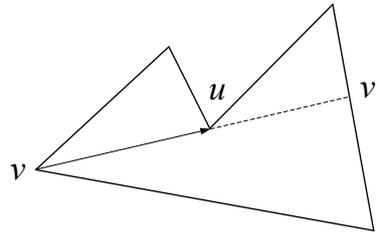


Figure 4. A back diagonal uv' created by a reflex vertex u and a vertex v visible from u .

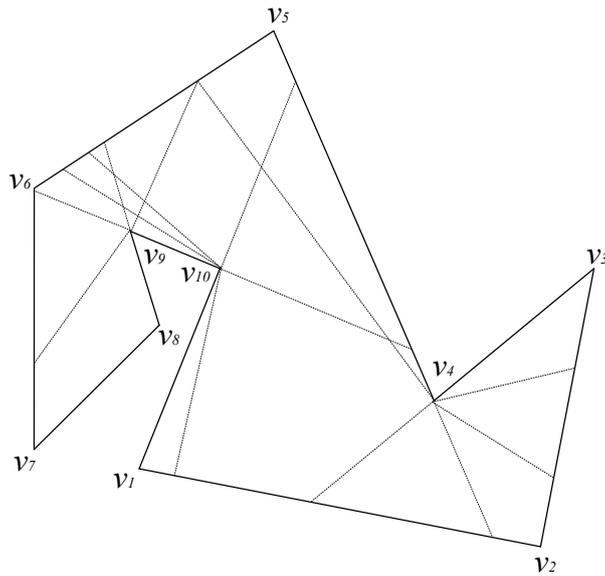


Figure 5. A back diagonal partitioning for a polygon with three reflex vertices.

3.3. Hidden Regions

Note that the gray areas in **Figure 3** are hidden regions with respect to the viewpoints p . We define *back edges* for two cases. *Case 1:* If p' is an interior point of an edge e of P , then p' is said to be a *reflection point* of p on e through the active reflex vertex s , and e is called a *back edge* of p through s . The edge v_7v_8 in **Figure 3** is the back edge of the viewpoint p through the reflex vertex s . *Case 2:* If p' coincides with a vertex v_i of P , then the back edge is taken as $e = (v_i, v_{i+1})$. In this case, v_i and s induce a *back diagonal* on which p lies. For ease of exposition, all of the following will be in terms of case 1. Note, that case 1 corresponds to a point p that does not lie on a back diagonal with respect to s . A *left hand hidden region* $H(L)$ with respect to p , a left hand active reflex vertex s , a reflection point p' and its back edge $v_{i-1}v_i$, is

a region bounded by the polygon defined by the circular vertex list:

$$H(L) = H(L, p, s) = (p', v_i, v_{i+1}, v_{i+2}, \dots, v_k, s, p')$$

where $v_i, v_{i+1}, v_{i+2}, \dots, v_k$, and s are the original vertices of P in CCW order. A *right hand hidden region* $H(R)$ is similarly defined for a right hand reflex vertex as $H(R) = H(R, p, s) = (p', s, v_i, v_{i+1}, v_{i+2}, \dots, v_k, p')$. **Figure 3(b)** provides an illustration of $H(L)$ and $H(R)$ for a given observation point p .

We can now define a visible polygon in terms of the definitions given above. Given a polygon P and an observation point p , define the visible polygon $V(p)$ as the polygon comprised of subsequences of the original vertices of P and all *windows* with respect to p . The visible polygon $V(p)$ contains all visible vertices and the reflection points of all active reflex vertices with respect to p . From proposition 2 it should be clear that the regions enclosed by $H(L)$ and $H(R)$ are not visible from p .

3.4. Calculation Formulas Using Hidden Areas

Define a rectangular coordinate system in E^2 such that each point i in E^2 is represented by a set of ordered real valued coordinates (x_i, y_i) . Given the coordinates of an observation point $p = (x_p, y_p)$, the end points of back edge (v_1, v_2) , and an active reflex point s , the reflection point p' can be calculated as a function of p and has the general form:

$$(x_{p'}, y_{p'}) = \left(\frac{\rho_1 x_p + \rho_2 y_p + \rho_3}{\rho_4 x_p + \rho_5 y_p + \rho_6}, \frac{\rho_7 x_p + \rho_8 y_p + \rho_9}{\rho_{10} x_p + \rho_{11} y_p + \rho_{12}} \right)$$

Here, ρ_i ($i = 1, \dots, 12$) are constants in terms of the points v_1, v_2 and s . The *hidden area* as a function of p' (the reflection point) has the following form:

$$A(H(L)) = \alpha_1 x_{p'} + \alpha_2 y_{p'} + \alpha_3, \quad A(H(R)) = \alpha_4 x_{p'} + \alpha_5 y_{p'} + \alpha_6$$

Here, α_i ($i = 1, \dots, 6$) are constants comprised of the coordinates of the original boundary vertices of P .

This is a direct result of applying Equation (2) to the boundary vertices $(p', v_i, v_{i+1}, v_{i+2}, \dots, v_k, s, p')$, $(p', s, v_i, v_{i+1}, v_{i+2}, \dots, v_k, p')$ of the left and right hidden regions, respectively. Given a point p in P , associate with it a set of active reflex vertices. Let $m_L(p)$ and $m_R(p)$ represent the number of left and right reflex vertices with respect to p . Each active reflex vertex i has an associated hidden region. Let $H(L, p, i)/H(R, p, i)$ represent a left/ right hidden regions of p associated with an active reflex vertex i , respectively.

The collection of left and right hand hidden regions associated with p are denoted as;

$S_L(p) = \{H(L, p, i), i = 1, 2, \dots, m_L(p)\}$ and $S_R(p) = \{H(R, p, i), i = 1, 2, \dots, m_R(p)\}$, respectively. Since intersections of hidden regions are empty (except for possible vertex points), the area of the visible polygon may be represented as follows:

$$A[V(p, P)] = A(P) - \left(\sum_{i=1}^{m_L(p)} A[H(L, p, i)] + \sum_{i=1}^{m_R(p)} A[H(R, p, i)] \right)$$

This area function will be derived in terms of the rectangular (x_p, y_p) coordinates of point p , in order to determine the directional move of p , such as to increase the visible area. For example, **Figure 6** shows a poly-

gon $P = \{M_1, M_2, \dots, M_k, v_1, v_2, \dots, s, M_{k+4}, M_{k+5}, \dots, M_n\}$, with a $H(L, p, s)$, and a visible polygon

$$V(p, P) = \{M_1, M_2, \dots, M_k, v_1, p', s, M_{k+4}, M_{k+5}, \dots, M_n\}.$$

Let $AL = A(P) - A(H(L, p, s))$ represent the area of the visible polygon associated with the view point p and a left hand hidden region $H(L, p, s)$. Using (2) and isolating the coordinates $(x_{p'}, y_{p'})$ we get the area in terms of p' :

$$2AL(p') = -(y_{v(1)} - y_s) \cdot x_{p'} + (x_{v(1)} - x_s) \cdot y_{p'} + \text{Const}$$

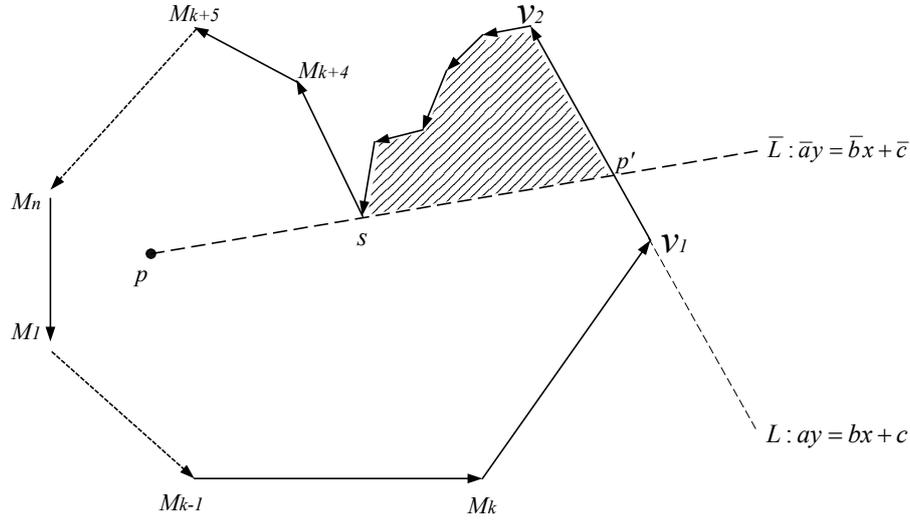


Figure 6. A simple polygon with a left hand hidden region.

$$\text{Const} = \sum_{i=1, \dots, k, k+4, \dots, n} x_{M(i)} \cdot (y_{M(i+1)} - y_{M(i-1)}) - x_{v(1)} \cdot y_{M(k)} + x_s \cdot y_{M(k+4)}$$

The only relevant vertices are the reflex vertex s (which created the hidden region) and v_1 which is the visible vertex of the edge $e = (v_1, v_2)$, the back edge of p through s . Note that p' is the intersection point between the line collinear with (p, s) and e . Repeating the process for a right hand side region, where v_2 is the visible vertex of the back edge e we obtain:

$$2AR(p') = (y_{v(2)} - y_s) \cdot x_{p'} - (x_{v(2)} - x_s) \cdot y_{p'} + \text{Const}$$

The individual coordinates of p' can be convex or concave, increasing or decreasing or constant in x_p and y_p depending on the orientation of the ray ps and the back edge (v_1, v_2) .

4. Gradient Search Algorithm for Maximal VAP

Following are the steps of a gradient search algorithm for finding a local max to the maximal visible polygon problem. Given the current point p , the algorithm finds the maximal gradient direction and moves p until one of a number of events occurs. These events are determined when the functional form of the area function undergoes a change if the move is continued. This occurs, for example, when the point hits a back diagonal or an original edge of P . Also, when the reflection point moves to an end point of its back edge, or when a new reflex point becomes active. At any of these events a new area function is introduced and the gradient is recalculated to determine a new movement direction.

4.1. Determination of the Maximal Gradient Direction

Since $x_{p'}$ and $y_{p'}$ depend on the view point p , we must rephrase the area's formulation using its coordinates. To do this two lines are constructed: $\bar{L} : \bar{a}y = \bar{b}x + \bar{c}$ (connecting p and s) and $L : ay = bx + c$ (collinear with the back edge (v_1, v_2)) (see **Figure 5**). The intersection of \bar{L} and L is the point $(x_{p'}, y_{p'})$. The right side of (9) contains functions of p' .

$$(x_{p'}, y_{p'}) = \left(\frac{\bar{a}c - a\bar{c}}{-\bar{a}b + ab}, \frac{-\bar{c}b + c\bar{b}}{-\bar{a}b + ab} \right)$$

$$\left(\frac{\partial(AL)}{\partial x_p}, \frac{\partial(AL)}{\partial y_p} \right) = \left(\bar{b} \cdot \left[\frac{ay_s - bx_s - c}{-\bar{a}b + ab} \right]^2, -\bar{a} \cdot \left[\frac{ay_s - bx_s - c}{-\bar{a}b + ab} \right]^2 \right)$$

$$\left(\frac{\partial(AR)}{\partial x_p}, \frac{\partial(AR)}{\partial y_p} \right) = \left(-\bar{b} \cdot \left[\frac{ay_s - bx_s - c}{-\bar{a}b + \bar{a}\bar{b}} \right]^2, \bar{a} \cdot \left[\frac{ay_s - bx_s - c}{-\bar{a}b + \bar{a}\bar{b}} \right]^2 \right)$$

The total gradient ∇_p is the sum of the partials for all active reflex vertices with respect to p .

$$\nabla_p = (\nabla_x, \nabla_y) = \left(\sum_{s \in S_L(p)} \frac{\partial(AL)}{\partial x_p} + \sum_{s \in S_R(p)} \frac{\partial(AR)}{\partial x_p}, \sum_{s \in S_L(p)} \frac{\partial(AL)}{\partial y_p} + \sum_{s \in S_R(p)} \frac{\partial(AR)}{\partial y_p} \right)$$

where, $S_L(p)$ and $S_R(p)$ are all active left and right reflex vertices with respect to p , respectively.

4.2. Max VAP Algorithm (the Gradient Search Algorithm)

Although this algorithm is based on movements between sub regions of a BDP, there is no need to actually construct it. It is only necessary to construct the back diagonals (line equations). This is because the algorithm searches for a critical event when the observer crosses a back diagonal. At his point, based on proposition 4, a recalculation of the area and new gradient are necessary.

The Steps of the Procedure

Given a simple polygon, and a point p^0 in P . Set $p = p^0$.

Step 1 Compute the area of P as $A(P)$.

Step 2 Determine the visible polygon $V(p)$ from p and its area $A(V(p)) \leq A(P)$. Identify all ‘‘active’’ reflex vertices s_1, s_2, \dots, s_w and the corresponding set of windows: $W(p) = \{W_k = (s_k, p'_k) \mid k = 1, \dots, w\}$, where s_k is the k^{th} active reflex vertex, and p'_k its reflection point which lies on the interior or endpoint its associated back edge $e_i^k = [v_i^k, v_{i+1}^k]$. If $W(p) = \phi$, go to Step 7 (Stop).

Step 3 Find the total gradient ∇_p that maximizes the increase in $A(V(p))$, using (12) and Proposition 5. If no such gradient exists, stop and go to Step 6.

Step 4 If p lies on an original edge of P , calculate a new gradient direction so as to slide along the edge in a direction such that the angle between the original gradient direction and the edge is minimal.

Step 5 Move p in the gradient direction until one of the following critical events occurs:

- p hits a vertex or edge of P .
- One of the reflection points p'_k slides to the endpoint of its back edge e_i^k (i.e. one of the vertices of P).
- If one of the windows W_k becomes collinear with one of the edges that forms a reflex vertex, i.e.; (s_k, s_{k-1}) or (s_k, s_{k+1}) for a *LH* or *RH* window, respectively. This reflex vertex becomes inactive; drop its associated window from $W(p)$.
- A new reflex vertex becomes active. Add the associated window to $W(p)$.

Step 6 If $W \neq \phi$, compute the new point p , and return to Step 2; otherwise, $W = \phi$, and the entire polygon is viewable from p .

Step 7 Stop. No local improvement is possible. Let p^* be the best point. Compute $A(V(p^*))$, the total visible area from p^* .

Note that the events in Step 5 correspond to the intersection of the gradient direction vector and all back diagonals.

Proposition 5 The sequence $[A(V(p^0)), A(V(p^1)), \dots, A(V(p^i)), \dots, A(V(p^*))]$ is increasing in i .

Proposition 6 The Gradient Search Algorithm converges in a finite number of steps. This is true since the sequence in Proposition 1 is bounded from above by the area of P .

Proposition 7 The gradient search algorithm has a time complexity $O(r^2 n^2)$.

Proof. In Step 2, the initial polygon can be calculated in time $O(n)$. This also identifies all active reflex vertices (bounded by r). Finding ∇_p will requires at r most calculations. In Step 3, the movement in the direction of ∇_p is performed until hitting a boundary edge or back diagonal. This requires $r \cdot n$ calculations to find all intersections between the gradient directional line and all back diagonal lines. Steps 2 and 3 are repeated at most $r \cdot n$ times in moving from back diagonal to back diagonal (resulting in at most $r \cdot n$ points between

p^0 and p^* as back diagonals are not revisited. Thus, the total time complexity of the gradient search algorithm is $O(n) + [r + rn] \cdot rn \approx O(r^2 n^2)$.

4.3. Illustrative Example for the Maximal VAP Problem

The example in **Figure 7** provides a demonstration of the algorithm for the case of a simple polygon with three reflex vertices, one of them, v_4 remaining inactive throughout. The sequence of points and visible areas are $p^0 \rightarrow p^1 \rightarrow p^2 \rightarrow p^*$, and $A^0 = 86.5 \rightarrow A^1 = 87.0 \rightarrow A^2 = 95.5 \rightarrow A^* = 100$.

5. Application-Solving the Guardhouse Problem

The guardhouse problem, proven to be NP-Hard in [4], is defined as the problem of the finding the minimal number of observation points F in a given a simple polygon P such that, the union of the visible area of all points in F equals P .

$$\text{Min} \{|F|, F \subseteq P\}, \quad \text{subject to: } \bigcup_{p \in F} V(p, P) = P$$

A *greedy algorithm* is described for solving the guardhouse problem which repeatedly uses the maximal VAP algorithm. The idea is to start with a simple polygon, and remove a maximal VAP. This operation can result in disconnected components, each in itself a simple polygon. The process is repeated for any remaining components. Each removal results in an additional observation point, so that the number of removals equals the number of points needed to view the entire original polygon.

5.1. The Greedy Heuristic

Given a simple polygon P , let Π be a collection of polygons.

- 1) Initially set $\Pi = \{P\}$, remove P from Π and set it to the current polygon P .
- 2) Start with a random point p in P . Call the Max VAP gradient search algorithm to find the point p^* , $V(p^*, P)$ and $A(V(p^*, P))$.
- 3) Let \bar{P} be the set of simple polygons obtained by subtracting the visibility polygon $V(p^*, P)$ from P . (i.e. the parts of the polygon P which are not visible from p^*). Add \bar{P} to the collection of polygons Π .
- 4) If $\Pi = \emptyset$ stop, otherwise;
- 5) Select a polygon P in Π and remove it, and return to Step 2.

The greedy heuristic described above terminates after a finite number of steps. The complexity of the greedy heuristic for solving the guardhouse problem is $O(r^3 n^2)$. There are at most r visits to Step 2 each requiring a call to the max VAP algorithm. Thus, the time complexity of the greedy heuristic is $r \cdot O(r^2 n^2) \approx O(r^3 n^2)$.

5.2 Illustrative Example

Figure 8(a) shows a sample polygon containing 16 vertices. We start with point shown in (b) and improve it until reaching a local maximum. The visible area is then removed from the initial polygon, and two new sub polygons appear in (c). In (d) and (e), this process is repeated with the two parts left, resulting in the final solution (f) in which the entire polygon is covered by 4 observation points.

6. Conclusions and Future Work

This work defined the maximal VAP problem, and offered a gradient algorithm for its solution which had a time complexity of $O(r^2 n^2)$. In most practical environments, r is much smaller than n , such that the complexity can be in fact much closer to $O(n^2)$. The algorithm was used as a basis for a greedy heuristic procedure for solving the guardhouse problem with a time complexity of $O(r^3 n^2)$. This heuristic procedure can be used for various applications in the field of homeland security such as placing indoor surveillance cameras or motion sensors. Several small examples are presented to illustrate the procedure.

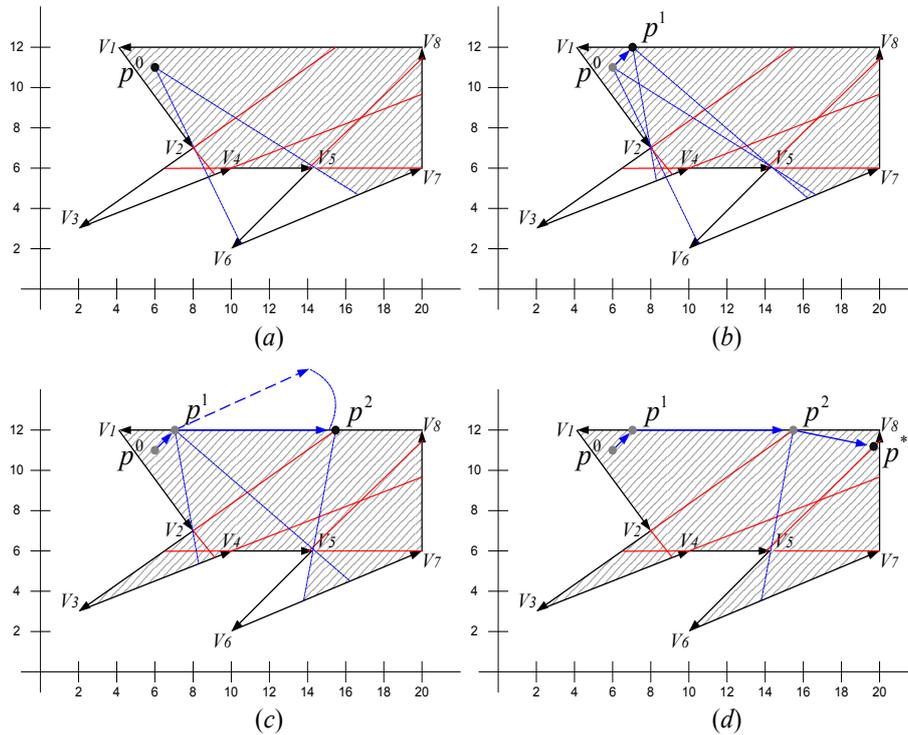


Figure 7. Steps of the max VAP algorithm starting with an initial view point p^0 .

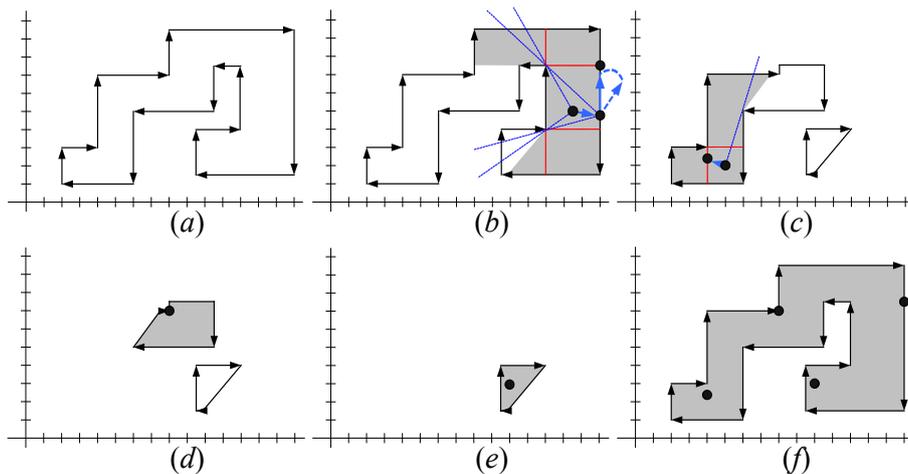


Figure 8. Example solution to the guardhouse problem using the greedy VAP heuristic algorithm.

It will be interesting to compare the results of our suggested procedure to other well-known heuristics for the guardhouse problem offered by other researchers. Such a comparison must include not only the quality of the solution, but also the running times. This is left for future work.

References

- [1] El Gindy, H. and Davis, D. (1981) A Linear Algorithm for Computing the Visible Polygon from a Point. *Journal of Algorithms*, **2**, 186-197. [http://dx.doi.org/10.1016/0196-6774\(81\)90019-5](http://dx.doi.org/10.1016/0196-6774(81)90019-5)
- [2] Stern, H. (1989) Polygonal Entropy: A Convexity Measure. *Pattern Recognition Letters*, **10**, 229-235. [http://dx.doi.org/10.1016/0167-8655\(89\)90093-7](http://dx.doi.org/10.1016/0167-8655(89)90093-7)
- [3] Cheong, O., Efrat, A. and Har-Peled, S. (2004) On Finding a Guard That Sees Most and a Shop That Sells Most. *Proc.*

15th ACM-SIAM Sympos. *Discrete Algorithms* 1098-1107.

- [4] O'Rourke, J. (1987) *Art Gallery Theorems and Algorithms*. Oxford University Press, Oxford.
- [5] Amit, Y., Mitchell, J.S.B. and Packer, E. (2010) Locating Guards for Visibility Coverage of Polygons. *International Journal of Computational Geometry & Applications*, **20**, 601-630. <http://dx.doi.org/10.1142/S0218195910003451>
- [6] Lee, D.T. and Lin, A.K. (1986) Computational Complexity of Art Gallery Problems. *IEEE Transactions on Information Theory*, **32**, 276-282. <http://dx.doi.org/10.1109/TIT.1986.1057165>