2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007



# Barrier Coverage for Variable Bounded-Range Line-of-Sight Guards

**Stephen Kloder** **Seth Hutchinson**
The Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
Urbana, IL 61801
Email: {kloder, seth}@uiuc.edu

*Abstract*— **In this paper, we formalize the problem of barrier coverage, that is, the problem of preventing undetected intrusion in a particular region using robot sensors. We solve the problem of finding the minimum-length barrier in the case of variable bounded-range line-of-sight sensors in a two-dimensional polygonally-bounded region. We do this by building a graph of candidate barriers that could potentially be in the minimum barrier. The dual of this graph shows the connectivity of the free space. We thus reduce the problem to the network flows maximum-flow/minimum-cut problem.**

## I. INTRODUCTION

In this paper, we address the problem of barrier coverage, the problem of preventing an intruder from entering a specific area without being seen by a "guard." Barrier coverage has applications in military security, private security, and sensor networks [1], [2]. Specifically, we are looking at the problem of protecting a region in the plane using variable bounded-range line-of-sight detectors that can detect an intruder crossing its line of sight, but only within a certain range. This range is not a fixed parameter, but is set by the deployer. We call these sensors segment guards.

Barrier coverage is one of the three types of coverage defined by Gage [3]. The other two are *blanket coverage* and *sweep coverage*. The goal of blanket coverage [4], [5], [6] is to maximize the total area the robots can see. The goal of sweep coverage [7], [8], [9] is to ensure that every point in a region is seen by some robot as the robots move across it. This problem can also be a single-robot problem [10], [11], [12].

In sensor network literature, barriers are usually generated for rectangular regions, which have been preselected as moats around the protected area. In [1] the region can also be annulus-shaped, but the structure is the same. A variety of methods have been used to generate barriers: [13] uses potential fields, [14] uses incremental random deployments, and [1] generates a grid of sensors. These approaches are related to the problem of *minimum exposure path* [15], [16], the problem of finding the path which is least likely to be seen by a sensor.

Barrier Coverage is related to several separation problems from computational geometry. There are several problems of separating polygons into separate regions, either with line segments [17], circles [18], wedges [19], or strips [19]. All of these assume polygons in open space. Furthermore, [20] looks at separating point sets inside polygons using chords.

Consider a mobile entity (a robot, a person, etc.), which we call an *intruder*. The intruder is known to be somewhere inside a *start region* $\mathcal{S}_1$, and will try to enter some *stop region* $\mathcal{S}_2$. A *barrier* is a set of robot configurations that prevents the intruder from entering $\mathcal{S}_2$ without being seen by at least one robot. We call these robots *guards*, but they can be viewed as robots with sensors or as fences. Anything that prevents undetected intrusion can serve as a guard.

This idea of guards and intruders suggests two problems. The first is to determine whether a given set of guards is a barrier. The second is to find a minimum-cost barrier for a particular environment and a particular type of guard. In this paper we focus on two-dimensional polygonally-bounded environments, point intruder, and segment guards. The former problem can be solved just by checking the connectivity of the set of points the guards cannot see. The latter problem remains nontrivial even in this example, so we will focus on it. We give a solution using geometry and graph theory network flows.

In this paper we begin to generalize the problem of barrier coverage to a greater variety of environments than those in the aforementioned sensor networks approaches, particularly in the use of domains that are polygons with and without holes. This allows us to look at situations where the environment as well as intruder locations and goals can be more complicated. In this way we bring barrier coverage closer to the computational geometry separation problems.

This paper is laid out as follows: in Section II we formally define barrier coverage, in Section III we define a Barrier Candidate Graph, which we use to solve the barrier coverage problem in Section IV. In Section V we expand the method to cover less idealized problems. In Section VI we describe future work, and in Section VII we give concluding remarks.

## II. PROBLEM DEFINITION

In this paper we focus on a point intruder $(x_I, y_I) \in \mathbb{R}^2$ moving in the plane. The intruder can only be in the obstacle-free workspace $\mathcal{W} \subset \mathbb{R}^2$, which is compact, connected, and bounded by polygons. The intruder is known to originate somewhere in the *start set* $\mathcal{S}_1 \subset \mathcal{W}$, and is trying to travel to some point in the *stop set* $\mathcal{S}_2 \subset \mathcal{W}$. Both $\mathcal{S}_1$ and $\mathcal{S}_2$ are compact and bounded by polygons. Figure 1 shows an example problem domain.

Each guard $q_j$ can be written as $q_j = (x_j, y_j, \theta_j, r_j) \in \mathbb{R}^2 \times S^1 \times \mathbb{R}^+$. This guard is located at $(x_j, y_j)$ and can see

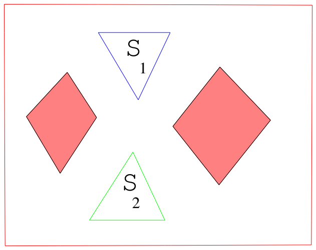



Fig. 1. Sample Barrier Problem Domain. The shaded regions are obstacles

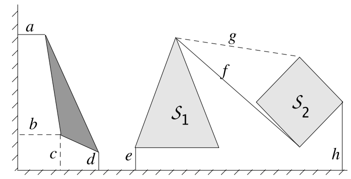



Fig. 2. Sample Barrier Candidates. The dashed lines are not included in the Barrier Candidate Graph

in direction $\theta_j$ up to a distance of its range, $r_i$. Guards are further restricted in that they must reside in the workspace (i.e. $(x_i, y_i) \in \mathcal{W}$, and that they cannot see through walls, i.e. past points not in $\mathcal{W}$.

For each guard $q = (x, y, \theta, r)$ we define a *visibility region* $V(q)$ to be the set of points that $q$ can see. Since $q$ can see a straight line until it goes outside of range or hits an obstacle, $V(q)$ is the maximal connected component of $\{(x + k\cos\theta, y + k\sin\theta) | 0 \le k \le r\} \cap \mathcal{W}$ that contains $(x, y)$. These guards are called segment guards because the visibility region of each valid guard is a line segment. For the remainder of this paper, when we describe a segment guard, we will describe it in terms of its visibility region, rather than in terms of its location and orientation.

A set of guards $\{q_1, \dots, q_n\}$ is a *barrier* iff every path from $\mathcal{S}_1$ to $\mathcal{S}_2$ in $\mathcal{W}$ intersects $V(q_j)$ for at least one $j$. This means that the intruder cannot get from its start to its intended goal without being seen by a guard. Equivalently, $\{q_1, \dots, q_n\}$ is a barrier iff $\mathcal{S}_1$ and $\mathcal{S}_2$ are in separate connected components of $\mathcal{W} - \bigcup_{j=1}^{n} V(q_j)$.

The goal here is to find the minimum-length barrier, i.e. the set of guards $\{q_1, \dots, q_n\}$ that is a barrier, and has minimum $\sum_{j=1}^{n} r_j$. This reflects situations where, for example, robots that see farther are more expensive to construct, or where long segment guards are composed of many smaller segment guards.

## III. Barrier Candidate Graph

Here we define a barrier candidate graph that contains edges that are relevant towards minimum barrier coverage. These are the barrier candidates. This graph is related to the reduced visibility graph [21].

If $f_1$ and $f_2$ are features (where a feature is an edge or a vertex), the shortest segment from a point in $f_1$ to a point in $f_2$ is called the *minimal segment* from $f_1$ to $f_2$. This segment is unique unless $f_1$ and $f_2$ are parallel edges.If $v$ is the vertex incident to edges $e_1$ and $e_2$ and the minimal segment from $e_1$ to $e_3$ has $v$ as an endpoint, the segment is *redundant* unless it is also the minimal segment from $e_2$ to $e_3$. In Figure 2, the edges labeled $a$, $b$, $c$, and $d$ are minimal segments. The segments $b$ and $c$ are redundant.

A segment $s$ is *tangent* to a polygon at a vertex $v$ if inside some neighborhood of $v$, the line through $s$ intersects the polygon at the boundary, but not the interior. A *bitangent* is a segment tangent to two polygons. This bitangent is *separating* if the two polygons are on opposite sides of the line through $s$, and *supporting* if they are on the same side. In Figure 2, $e$ and $h$ are tangents, $f$ is a separating bitangent, and $g$ is a supporting bitangent.

The Barrier Candidate Graph is composed of minimal segments, tangents, and bitangents. Additional restrictions are applied to these segments. A segment is *admissible* if (1) it lies entirely inside $\mathcal{W}$, (2) it contains no points in the interior of $\mathcal{S}_1$ or $\mathcal{S}_2$, and (3) it is not redundant. The reasons for these restrictions will be explained in the proof below.

The barrier candidate graph of $\mathcal{W}$, $\mathcal{S}_1$, and $\mathcal{S}_2$ consists of all admissible segments that are:

1) minimal segments between obstacle edges,
2) minimal segments from vertices of $\mathcal{S}_i$ to obstacle edges, that are tangent to $\mathcal{S}_i$, $i = 1, 2$,
3) separating bitangents between $\mathcal{S}_1$ and $\mathcal{S}_2$, or
4) supporting bitangents between different vertices of $\mathcal{S}_1$, or between different vertices of $\mathcal{S}_2$,

Figure 3 shows the Barrier Candidate Graph for the example domain in Figure 1.

The barrier candidate graph is useful in finding minimum barriers, as it contains all possible segments of a minimum barrier, as we now demonstrate.

*Theorem 1:* In a workspace where no two obstacle segments are parallel, the minimum segment barrier separating $\mathcal{S}_1$ from $\mathcal{S}_2$ consists only of segments from the barrier candidate graph.

*Proof:*

We show that any barrier consists entirely of segments that are either in the barrier candidate graph, or can be shortened while maintaining the barrier. A barrier that contains segments of the latter type clearly cannot be a minimum barrier.

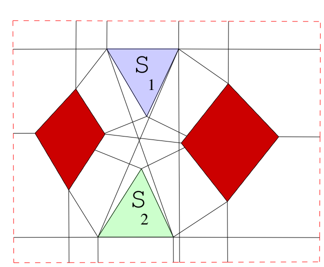
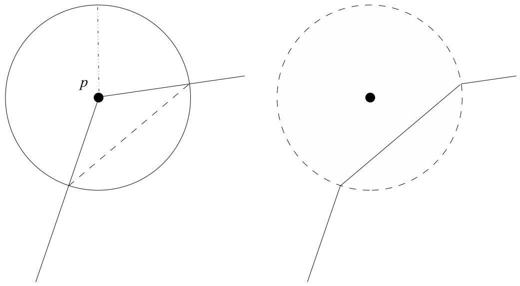



Fig. 3. Barrier Candidate Graph. The obstacle edges are dashed, and the barrier candidates are solid.




(a) Before (b) After

Fig. 4. Shortening by replacing an internal vertex

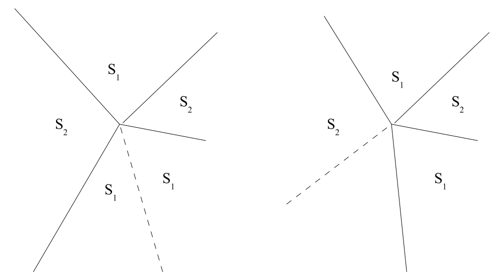
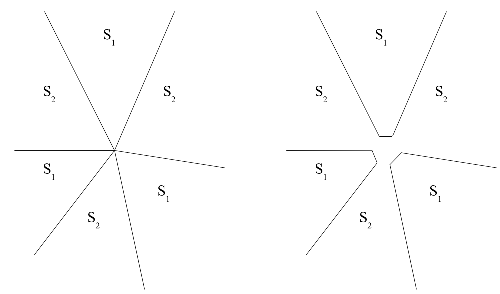



Fig. 5. Shortening by removing an unnecessary segment. In both cases, the dashed segment can be removed




(a) Before (b) After

Fig. 6. Shortening by combining multiple regions.

Therefore the minimum barrier must contain only segments of the former type.

The segments that can be used to construct a barrier can be classified according to their endpoints. A segment can have

1) at least one endpoint in the interior of $\mathcal{W} - \mathcal{S}_1 - \mathcal{S}_2$,
2) both endpoints on obstacle edges,
3) one endpoint at an obstacle edge, and one in $\mathcal{S}_1$ or $\mathcal{S}_2$, or
4) both endpoints in $\mathcal{S}_1 \cup \mathcal{S}_2$.

We show for each of these cases, either the segment is present in the barrier candidate graph, or can be shortened while preserving the barrier. Note that in all cases, any segments containing points in the interior of $\mathcal{S}_1$ or $\mathcal{S}_2$ (inadmissible segments) can be shortened by removing all such points, without changing the barrier.

1) If the segment has an endpoint $p$ in the interior of $\mathcal{W} - \mathcal{S}_1 - \mathcal{S}_2$, the barrier can always be shortened. The method of shortening depends on the degree of $p$, i.e. the number of segments incident to it.
   If the degree is one, the intruder can simply move around this segment. Therefore the segment does not contribute to the barrier, and can be removed completely. If the degree is 2, select $\epsilon > 0$ such that the disk radius $\epsilon$ centered at $p$ lies entirely on the interior of $\mathcal{W} - \mathcal{S}_1 - \mathcal{S}_2$. Replace the portion of the barrier inside the disk with a line segment connecting the two points on the disk's boundary. By the triangle inequality, this barrier is shorter. Note that this assumes the angle between these two edges is not $\pi$. Otherwise, the two segments can be combined into a single segment. See Figure 4.
   If the degree is 3 or greater, consider the regions that the complete barrier separates $\mathcal{W}$ into. Each region can be labeled by whether it contains points from $\mathcal{S}_1$, points from $\mathcal{S}_2$, or neither (if it contains points from both, it is not a barrier). Now, consider the regions that meet at this interior vertex. If there is a region with a "neither" label, or two adjacent regions with the same label, a segment can be removed. See Figure 5. Otherwise, pick two or more regions with the same label, and combine them with a method analogous to removing a degree-2 vertex. See Figure 6. The resulting barrier is shorter.
2) If a segment connecting two obstacle edges is the shortest possible segment connecting those two edges, it is a minimal segment. If it is not minimal, it can be shortened by moving one endpoint along the edge towards the minimal segment. See Figure 7. If it is a redundant minimal segment, then it can be shortened by moving it along the pair of edges for which it is not the minimal segment. If it is a non-redundant minimal segment, it is present in the barrier candidate graph. If this segment's interior contains an obstacle boundary point, the segment can be split into two segments of this type, each of which can be shortened the same way as before. If this segment contains a point in $\mathcal{S}_1$ or $\mathcal{S}_2$, then it can be split into two separate segments that are dealt with in the next item. Therefore, if a minimum barrier contains a segment that connects two obstacle edges, it must be a minimal segment. This is present in the barrier candidate graph (admissible segment type 1).
3) A segment with an endpoint, $v_1$, on an obstacle edge and the other, $v_2$, in $\mathcal{S}_1$ or $\mathcal{S}_2$ can be shortened unless it is the shortest possible segment between the obstacle edge and $v_2$. The method is the same as for the previous item. If a vertex is in the interior of $\mathcal{S}_1$, it can be shortened by removing the portion of the segment inside $\mathcal{S}_1$. The same is true for $\mathcal{S}_2$. Furthermore, if the segment is not a tangent, it cannot separate on its own, and requires another segment in the barrier. The two segments combined can be shortened in a way analogous to removing an interior vertex (the second

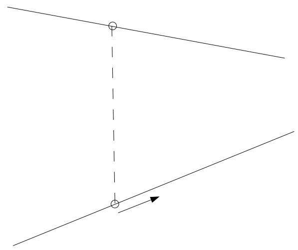
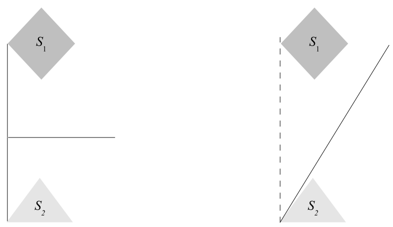


Fig. 7. Shortening a non-minimal segment. The solid lines are obstacle segments, and the dashed line is a candidate barrier. Moving the endpoint along the obstacle boundary in the direction of the arrow shortens the barrier.



(a) (b)

Fig. 8. Shortening a supporting bitangent: (a) is like Figure 5; in (b) the dashed segment can be removed without losing the barrier.

case in item 1). Therefore, a segment of this type must be a minimal segment and tangent to $\mathcal{S}_1$ or $\mathcal{S}_2$ in order to be part of a minimum barrier. This is present in the barrier graph (admissible segment type 2).

4) A segment connecting two points in $\mathcal{S}_1 \cup \mathcal{S}_2$ can be shortened unless it is a bitangent, using the method described in the previous item. If the segment is a supporting bitangent between $\mathcal{S}_1$ and $\mathcal{S}_2$, then it cannot separate on its own, and requires another segment. This new segment touches either an endpoint or the middle of the bitangent. If it touches the middle, it produces a degree-3 interior vertex, which is covered above. If it touches an endpoint, the bitangent can be removed altogether. See Figure 8. If the segment is a separating bitangent between components of $\mathcal{S}_1$, or between components of $\mathcal{S}_2$, then it has no effect (Separating components of $\mathcal{S}_1$ does not change the set of locations that can be reached from $\mathcal{S}_1$; it only changes where one can go from each component). Therefore such a bitangent can be removed. Therefore only separating bitangents between $\mathcal{S}_1$ and $\mathcal{S}_2$, supporting bitangents at $\mathcal{S}_1$, and supporting bitangents at $\mathcal{S}_2$ can be in minimum barriers. These all appear in the barrier candidate graph (admissible segment types 3 and 4).

If there exists a minimum barrier that contains segments not in the barrier candidate graph, then this barrier can be shortened using the methods described above. Therefore it is not minimum; this is a contradiction. Therefore, the minimum barrier must consist of segments from the Barrier Candidate Graph. ■

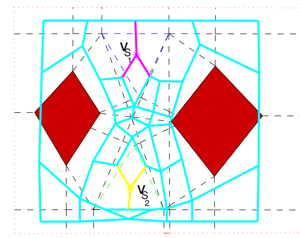


Fig. 9. Dual of Barrier Candidate Graph: Connectivity Network. Solid edges are in the network, dahsed edges are barrier candidates, and dotted lines are boundaries.

## IV. Barrier Candidate Graph Network Flows Approach

We construct a new network from the barrier candidate graph. To prepare this network, first remove any redundancies (e.g. two pairs of edges with the same minimal segment) from the barrier candidate graph. Add an edge for each obstacle boundary segment. Place vertices at every edge intersection, and subdivide the edges accordingly. The resulting graph is planar, so $\mathcal{W}$ can be decomposed into faces that are bounded by these subdivided edges.

Take the dual of this graph to produce a weighted graph as follows. Replace each face with a vertex. If two faces are adjacent, connect the corresponding new vertices with a new edge. This new edge corresponds to placing a barrier across the edge between the original faces, and is weighted according to the length of this edge. We call this graph the *connectivity network*, as the connectivity of the graph reflects the connectivity of the workspace. Traveling from one point in $\mathcal{W}$ to another point in a different region requires crossing some segments of the barrier candidate graph. This is equivalent to following a path of corresponding dual segments in the connectivity network. Similarly, separating $\mathcal{S}_1$ from $\mathcal{S}_2$ requires finding an edge cut in the connectivity network such that in the remaining graph there is no path from a vertex inside a component of $\mathcal{S}_1$ to a vertex inside a component of $\mathcal{S}_2$ in the remaining network.

Connect $v_{s_1}$ to all the vertices corresponding to regions that intersect $\mathcal{S}_1$, and connect $v_{s_2}$ to all the vertices corresponding to regions that intersect $\mathcal{S}_2$. These connections should have infinite weight (alternatively, combine all vertices inside $\mathcal{S}_1$ into $v_{s_1}$, and the equivalent for $v_{s_2}$). An edge cut that separates $v_{s_1}$ from $v_{s_2}$ corresponds to a barrier that separates $\mathcal{S}_1$ from $\mathcal{S}_2$. Figure 9 shows the network derived from the barrier candidate graph in Figure 3. The shaded regions are not regions and do not have vertices in the dual graph, as they are not inside $\mathcal{W}$.

The minimum edge cut separating $v_{s_1}$ from $v_{s_2}$ corresponds via the dual-graph relation to the shortest barrier that consists only of barrier candidate segments. According to Theorem 1 there are no shorter barriers. Therefore this barrier is the minimum barrier of any type, and must be the desired solution.

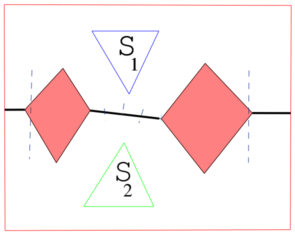



Fig. 10. Minimum Barrier. Dashed lines show corresponding edges in the dual graph

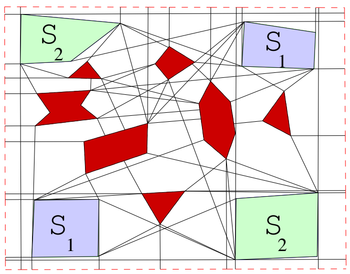



Fig. 11. Example Problem: Barrier Candidate Graph.

Therefore it suffices to solve the network flows min-cut problem, which is equivalent to the network flows max-flow problem [22]. This can be solved efficiently using augmenting paths [23] or preflows [24]. Figure 10 shows the minimum barrier of the sample domain. The dashed lines show the minimum cut for the connectivity graph.

This method scales well to more complex domains. Figures 11 and 12 show the algorithm applied to a domain with more obstacles and multiple components for $\mathcal{S}_1$ and $\mathcal{S}_2$. Figure 11 shows the workspace plus the barrier candidate graph, and Figure 12 shows the resulting minimum barrier, along with the network edges that it is contructed from. Furthermore, Figures 13 and 14 show the effects of removing one obstacle from this sample problem.

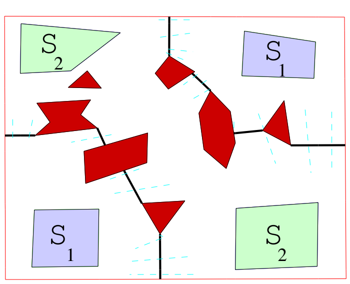



Fig. 12. Minimum Barrier. Dashed lines show corresponding edges in the dual graph

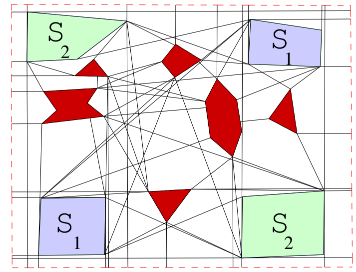



Fig. 13. Example problem less one obstacle: Barrier Candidate Graph.

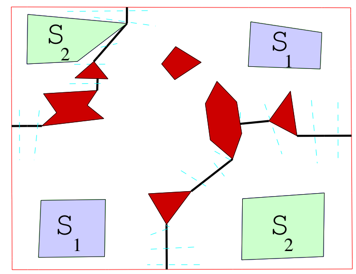



Fig. 14. Minimum Barrier of Workspace in Figure 13. Dashed lines show corresponding edges in the dual graph

## V. Accommodating Parallel Edges

In previous sections, we assumed no two obstacle edges were parallel. Here we show that our method can be adjusted to be applied to problem domains with parallel obstacle edges.

Parallel edges create a problem with the definition of the minimal segment. There are an infinite number of minimal segments between two parallel edges, all of with the same length. These are all the segments perpendicular to both obstacle edges. In this section we will show that with respect to barrier coverage, all such minimal segments are either (1) equivalent to each other, (2) equivalent to another segment already in the graph, or (3) not minimal (i.e. they can be shortened). Here we say two segments are equivalent if they have the same effect on the barrier. In the first case, all these minimal segments can be replaced by a single arbitrary edge. In the others, no edge needs to be included.

Let $e_1$ and $e_2$ be two parallel obstacle edges, and define $M = \{s\,|s$ is a minimal segment between $e_1$ and $e_2\}$. Now consider the components of $\mathcal{M} = \bigcup_{s \in M} s$ . All components are rectangles, and any two minimal segments in the same component are equivalent. Each component is bounded by a minimal segment that touches either (a) a vertex of $e_1$ or $e_2$, (b) a vertex of $\mathcal{S}_1$ or $\mathcal{S}_2$, or (c) an obstacle vertex that is not in $e_1$ or $e_2$.

If the component is bounded on both sides by segments of type (a), then $\mathcal{M}$ has one component, and all segments are equivalent. This is case (1) above; only one edge needs to be included. If the component is bounded on a side by a segment of type (b), this segment goes through a vertex $v'$ of $\mathcal{S}_1$ or

$\mathcal{S}_2$. This segment is the union of two segments: the tangent at $v'$ to $e_1$, and the tangent at $v'$ to $e_2$. Both of these segments are already in the barrier candidate graph (case (2) above). Similarly, if the component is bounded by a segment of type (c), it can be split into two obstacle-to-obstacle segments, at least one of which can be shortened further using the methods in the proof to Theorem 1. Therefore none of the segments in this component are minimal (case (3) above).

Therefore, a Barrier Candidate Graph can accommodate workspaces with parallel obstacle edges by including one arbitrary minimal segment for each pair of parallel obstacle edges. In these cases, while Theorem 1 still holds inasmuch as every minimum barrier consists of minimal segments or tangents, it is no longer true that every minimum barrier consists of segments from the barrier candidate graph. However, since we have shown that every minimal segment is equivalent to one in the barrier candidate graph, every minimum barrier is equivalent to a minimum barrier generated by searching the barrier candidate graph. This means the given method is applicable to enivronments with parallel obstacle edges.

## VI. Future Work

This paper shows barrier coverage in a specific problem domain. We would like to expand the definition to other problem domains.

We would like to look at different types of guards. This would include guards with fixed ranges, omnidirectional guards, guards in three dimensions, guards with placement constraints, and barriers composed of different types of guards. Similarly, we would like to look at different types of intruders. This would include intruders with volumes, changing shapes, and with kinematic and dynamic motion constraints.

We would also like to address barrier coverage when insufficent guards are available. The new goal is to minimize the intruder's probability of intruding. We call such a guard deployment a *partial barrier*, and we call its effectivelness in preventing intrusion *partial coverage*. The goal becomes to minimize probability of intrusion (i.e. maximize partial coverage) with a given set of guards.

## VII. Conclusion

In this paper we have formally defined barrier coverage, and shown how to find a minimum segment barrier in a two-dimensional polygonally-bounded region. We have shown that in these domains, the problem of finding the minimum-length segment barrier reduces to the two problems of finding barrier candidates, and finding the minimum edge cut. Both of these can be solved efficiently.

## References


[1] S. Kumar, T. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *Proceedings of the Eleventh Annual International Conference on Mobile Computing and Networking (ACM MobiCom)*, Cologne, Germany, 2005.

[2] O. Dousse, C. Tavoularis, and P. Thiran, "Delay of intrusion detection in wireless sensor networks," in *MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM Press, 2006, pp. 155–165.

[3] D. W. Gage, "Command control for many-robot systems," in *AUVS92, the Nineteenth Annual AUVS Technical Symposium*, Hunstville Alabama, USA, June 1992, pp. 22–24.

[4] S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," in *Proc. IEEE ICRA '04. Int. Conf. Robotics and Automation 2004*, vol. 1, New Orleans, LA, Apr. 2004, pp. 165 – 171.

[5] A. Howard, M. Mataric, and G. Sukhatme, "An incremental deployment algorithm for mobile robot teams," in *IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 3, Oct. 2002, pp. 2849 – 2854.

[6] R. Morlok and M. Gini, "Dispersing robots in an unknown environment," in *DARS 2004*, Toulouse, France, June 2004.

[7] D. Kurabayashi, J. Ota, T. Arai., and E. Yoshida, "Cooperative sweeping by multiple mobile robots," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 2, Apr. 1996, pp. 1744 – 1749.

[8] H. Choset, "Coverage for robotics - a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, Oct. 2001.

[9] X. Zheng, S. Jain, S. Koenig, and D. Kempe, "Multi-robot forest coverage," in *Proc. IROS 2005. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005*, Aug. 2005, pp. 3852–3857.

[10] A. Zelinsky, "A mobile robot exploration algorithm," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 707–717, 1992.

[11] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, pp. 247–253, 2000.

[12] S. Wong and B. MacDonald, "A topological coverage algorithm for mobile robots," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, Oct. 2003, pp. 1685–1690.

[13] S. Hynes and N. Rowe, "Multiagent simulation for assessing massive sensor deployment," *Journal of Battlefield Technology*, vol. 7, no. 2, pp. 23–36, July 2004.

[14] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Sept. 2002, pp. 42–48.

[15] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom '01)*, July 2001, pp. 139–150.

[16] G. Veltri, Q. Huang, G. Qu, and M. Potkonjak, "Minimal and maximal exposure path algorithms for wireless embedded sensor networks," in *ACM International Conf. on Embedded Networked Sensor Systems (SenSys)*, 2003, pp. 40–50.

[17] J. S. B. Mitchell and S. Suri, "Separation and approximation of polyhedral objects," *Computational Geometry*, vol. 5, pp. 95–114, 1995.

[18] J. D. Boissonnat, J. Czyzowicz, O. Devillers, and M. Yvinec, "Circular separability of polygons," *Algorithmica*, vol. 30, pp. 67–82, 2001.

[19] F. Hurtado, M. Noy, P. A. Ramos, and C. Seara, "Separating objects in the plane by wedges and strips," *Discrete Applied Mathematics*, vol. 109, pp. 109–138, 2001.

[20] E. D. Demaine, J. Erickson, F. Hurtado, J. I. S. Langerman, H. Meijer, M. Overmars, and S. Whitesides, "Separating point sets in polygonal environments," in *Proceedings of the 20th Annual ACM Symposium on Computational Geometry*, 2004.

[21] H. Rohnert, "Shortest paths in the plane with convex polygonal obstacles," *Information Processing Letters*, vol. 23, no. 2, pp. 71–76, Aug. 1986.

[22] L. R. F. Jr. and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[23] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton, NJ: Princeton U. Press, 1962.

[24] A. Karzanov, "Determining the maximal flow in a network by the method of preflows," *Soviet Mathematics Doklady*, vol. 15, pp. 434–437, 1974.