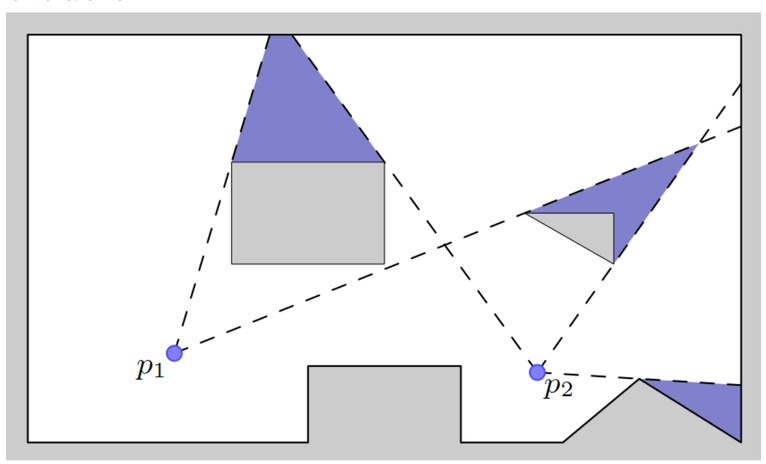
An Analytical Approach to Finding Critical Points in Multi-Robot Visibility-Based Pursuit-Evasion Problems

Ryan Lumbert,[1] Trevor Olsen,[2] and Jason M. O'Kane[2]

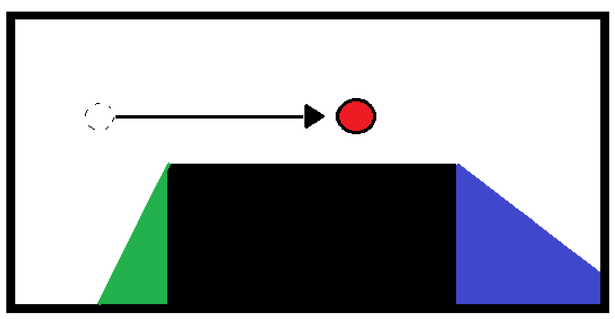
[1]Department of Computer Science, Winthrop University, Rock Hill, South Carolina
[2]Department of Computer Science and Engineering, University of South Carolina

Introduction

In the **multi-robot visibility-based pursuit-evasion problem**, a team of robots move through an environment to locate evaders.



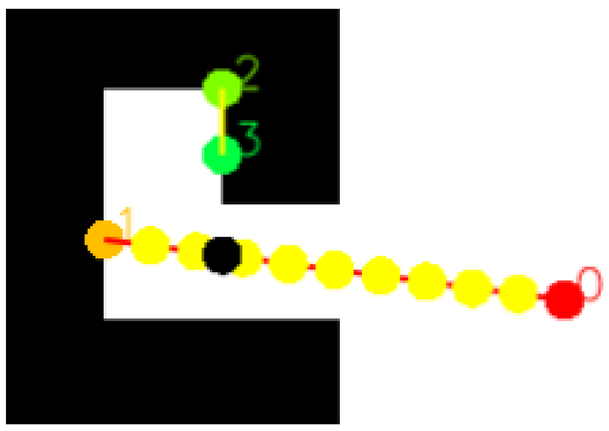
An example of two pursuers in an environment. The blue regions represent **shadow regions**, in which evaders may be hiding. The white regions represent the pursuers' **visibility regions**. [1].



The green shadow is cleared of evaders, while the blue region may contain pursuers, and is contaminated.

Objective

Current methods for tracking which shadow regions are contaminated sample frequently, which increases simulation times.



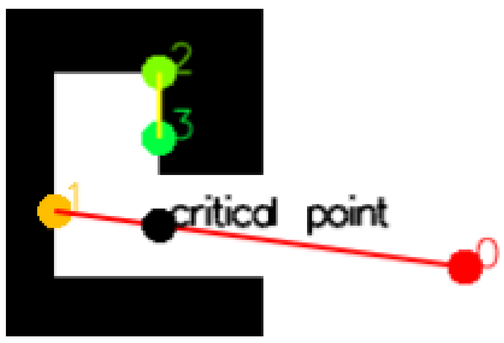
Goal: Decrease number of samples needed for simulation by computing **critical points**.

Problem Statement

It is desirable to only simulate moments at which a **shadow event occurs**.

- **Inputs:** An environment, E, and a list of pursuer configurations.
- **Output:** Critical points along the path of each pursuer where a shadow event occurs.

Type 1 Shadow Events

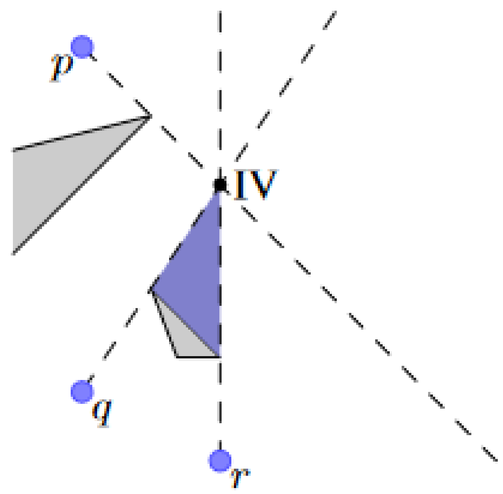


A Type 1 shadow event occurs when a pursuer is colinear with a pair of environment vertices.

Moment at which a critical event occurs for Type 1 shadow event:

$$t = \frac{c1x(p1y-c2y)+c1y(c2x-p1x)-c2xp1y+c2yp1x}{(c1x-c2x)(p1y-p2y)+c1y(p2x-p1x)+c2y(p1x-p2x)}$$

Type 3 Shadow Events

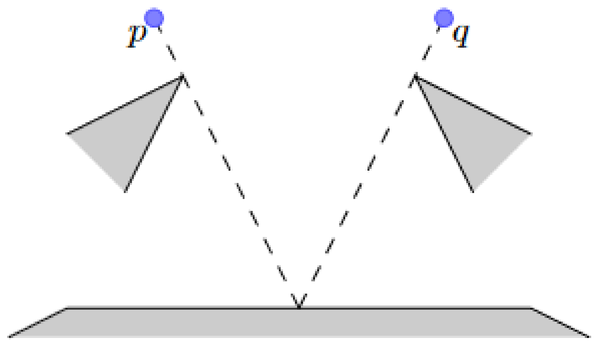


Type 3 shadow events occur when three occlusion rays meet a a single point [1].

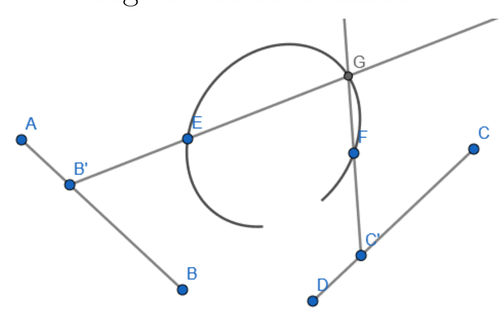
Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2050896.

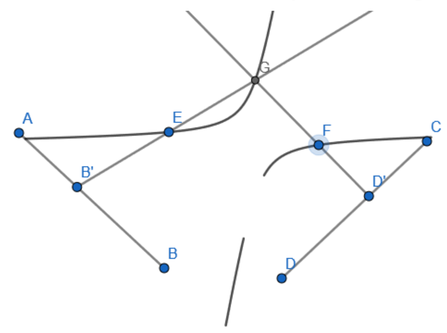
Type 2 Shadow Events



A Type 2 shadow events occurs when two occlusion rays intersect at an edge of the environment [1]. We discover where the critical points are by finding the conic traced by the intersection of the pursuers' occlusion rays and observing where that intersects an edge of the environment.



Intersection of occlusion rays tracing an ellipse.



Intersection of occlusion rays tracing a hyperbola.

Further Research

Further research is needed to show that this process will decrease simulation times.

References

[1] Nicholas M Stiffler and Jason M O'Kane. A complete algorithm for visibility-based pursuit-evasion with multiple pursuers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1660–1667. IEEE, 2014.