Recovering Visibility and Dodging Obstacles in Pursuit-Evasion Games
Ahmed Abdelkader
Department of Computer Science, University of Maryland at College Park

Introduction
Visibility-based pursuit-evasion:
- Motion: holonomic, max speed per player
- Visibility: omnidirectional, optional range
- Traditionally, the game ends when the pursuer loses sight of the evader.
- Applications:
  - Surveillance, monitoring, FPS and racing games, etc.

Discretization and Strategy Matrix
- Represent the map as a grid of pixels (white: clear, black: obstacle).
- For each pair of positions \((p, e)\), \(S[p, e] = 1\) iff the evader can win.
- Easily accommodates different motion and sensing models.
- Solve for the optimal strategy by backward induction.
- \(\mathcal{N}(x)\) denotes the neighboring locations player \(x\) can reach in 1 turn.

The Classical (Primal) Game
- Is it possible to keep the evader in sight? How?
- Initialize with visibility queries then solve the recurrence:
  \[
  S[p, e, i] = \begin{cases} 
  \neg v(p, e) & \text{if } i = 0, \\
  \bigvee_{e' \in \mathcal{N}(e)} \bigwedge_{p' \in \mathcal{N}(p)} S[p', e', i - 1] & \text{otherwise.}
  \end{cases}
  \]

Conclusions & Future Work
- Optimal strategies allowing visibility to be recovered within \(d\) turns
- Optimal strategies in dynamic environments
- Future directions:
  - State space reduction for improved running times
  - Generate optimal strategies for more than two players