Placement and Motion Planning Algorithms for Robotic Sensing Systems

Pratap Tokekar

Ph.D. Thesis Defense

Adviser: Prof. Volkan Isler

Data-Driven Science

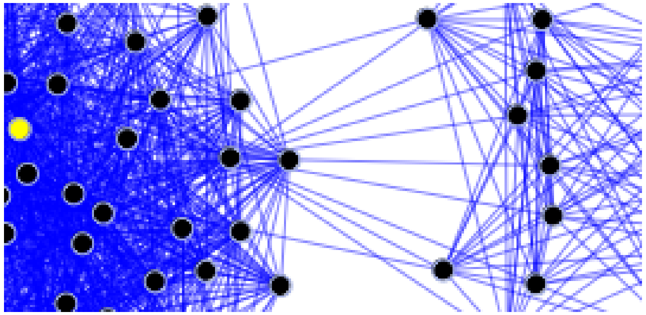
- Data for understanding complex phenomena
- Emergence of new data analytics techniques

Data-Driven Science

- Data for understanding complex phenomena
- Emergence of new data analytics techniques
- **Where does the data come from?**

Data-Driven Science

- Data for understanding complex phenomena
- Emergence of new data analytics techniques
- **Where does the data come from?**

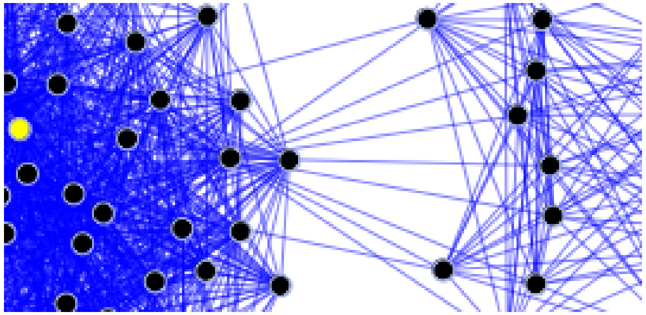


Source: wikipedia.org

Facebook, Twitter, Wikipedia, etc.

Data-Driven Science

- Data for understanding complex phenomena
- Emergence of new data analytics techniques
- **Where does the data come from?**



Source: wikipedia.org

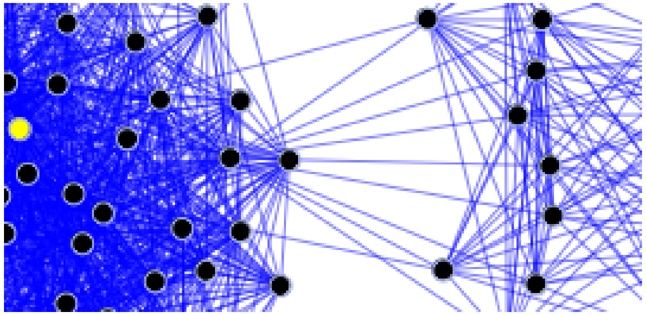
Facebook, Twitter, Wikipedia, etc.

Source cires.colorado.edu

Agriculture, Air Quality Monitoring, Migratory Birds Tracking

Data-Driven Science

- Data for understanding complex phenomena
- Emergence of new data analytics techniques
- **Where does the data come from?**



Source: wikipedia.org

Facebook, Twitter, Wikipedia, etc.

Source cires.colorado.edu

Agriculture, Air Quality Monitoring, Migratory Birds Tracking

- Manual data collection can be tedious, limited, and even infeasible at times

Ideal Job for Robotic Sensing Systems!

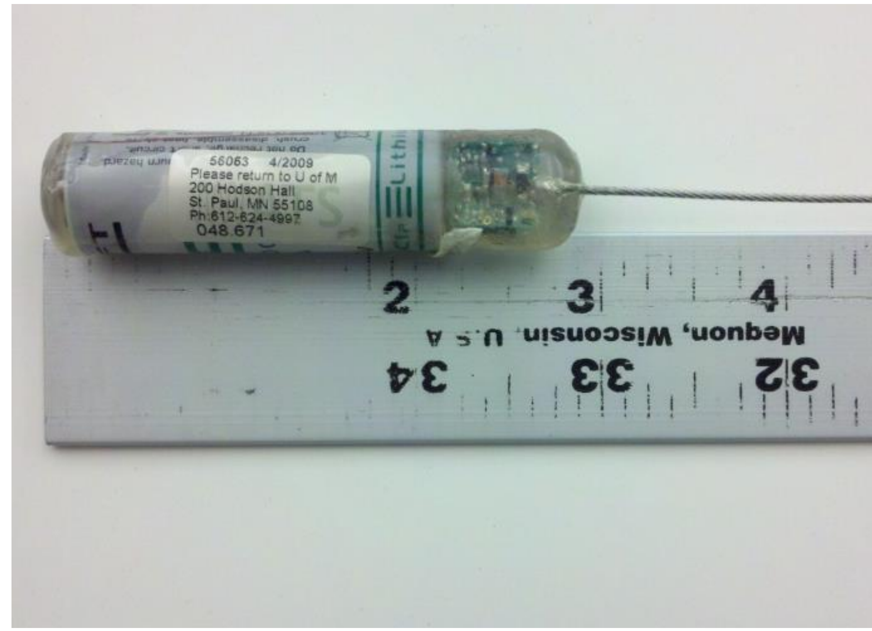
Research Challenges

- Robust System Design
- Sensing Planning Algorithms
- Long-term operation
- New Robot Mechanisms
- Multi-Robot Coordination
- …

Putting it all together

- In this thesis, we focus on the research challenge of planning for the sensing in robotic sensing systems

Monitoring Invasive Fish



Sorensen Lab, Dept. of Fisheries, University of Minnesota

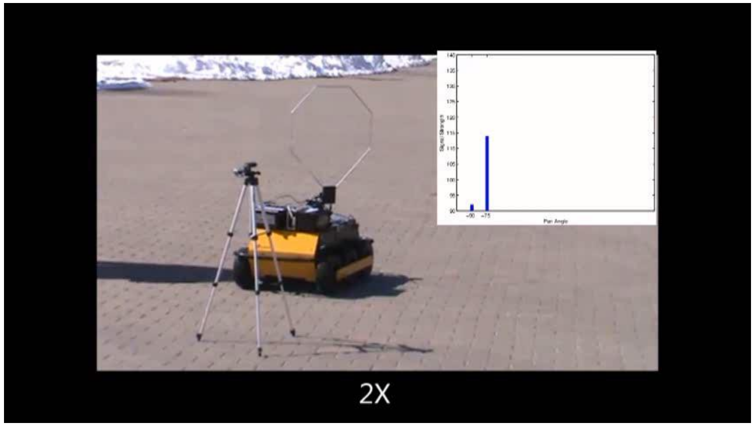
Tokekar, Bhadauria, Studenski and Isler. **A Robotic System for Monitoring Carp in Minnesota Lakes.** Journal of Field Robotics, 2010.

Sensing Tasks

1. Coverage: How to quickly cover the lake to search for the fish? Instead of covering the entire lake, biologists can input set of regions likely to contain fish.

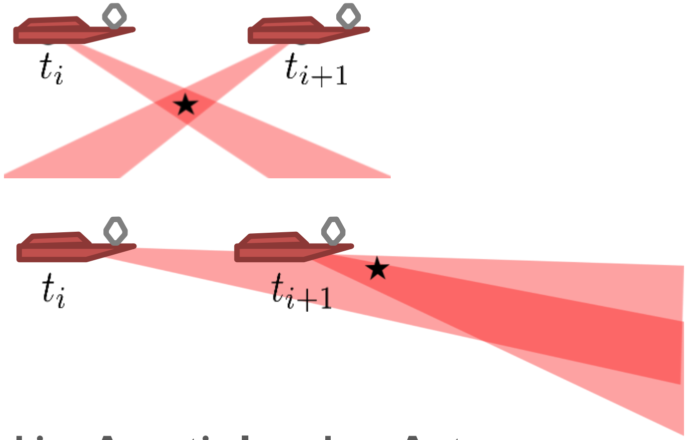
Tokekar, Branson, Vander Hook and Isler. **Tracking Aquatic Invaders: Autonomous Robots for Monitoring Invasive Fish.** IEEE Robotics & Automation Magazine, 2013.

Sensing Tasks



<u>1. Coverage:</u> How to quickly cover the lake to search for the fish? Instead of covering the entire lake, biologists can input set of regions likely to contain fish.

<u>2. Active Localization:</u> Locate the tags with **noisy** and **slow** bearing measurements. How to **adaptively** choose sensing locations?



Tokekar, Branson, Vander Hook and Isler. **Tracking Aquatic Invaders: Autonomous Robots for Monitoring Invasive Fish.** IEEE Robotics & Automation Magazine, 2013.

Thesis Contributions

- **Placement for Stationary Sensors**
  - Visibility-based Coverage with Orientation
    [ICRA '14]
  - Target Localization with Bearing Sensors
    [ICRA '13] [Submitted to T-ASE]

- Devise algorithms with theoretical performance guarantees
- Prototype system design
- Evaluate algorithms through field experiments

- **Motion Planning for Mobile Sensors**
  - Coverage and Active Localization with Robotic Boats: Carp Monitoring
    [ICRA '10, IROS '11] [JFR '10, R&A Magazine '13]
  - Sampling Algorithms for Ground & Aerial Robots: Precision Agriculture
    [IROS '13] [Submitted to T-RO]
  - Multi Target Tracking with Teams of Aerial Robots
    [IROS '14]
  - Energy-optimal Trajectory Planning
    [ICRA '11] [AURO '14]

Thesis Contributions

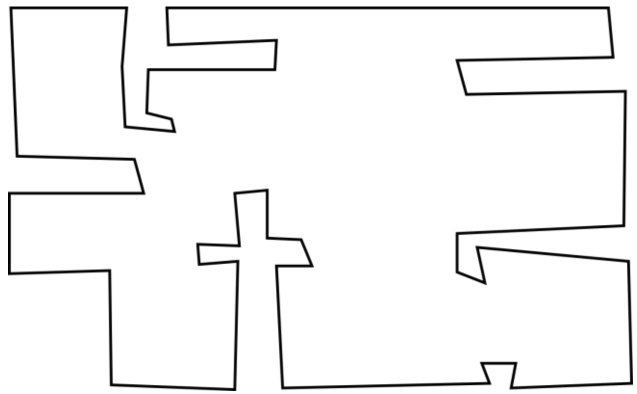
- **Placement for Stationary Sensors**
  - Visibility-based Coverage with Orientation
  - Target Localization with Bearing Sensors

- **Motion Planning for Mobile Sensors**
  - Coverage and Active Localization with Robotic Boats: Carp Monitoring
  - Sampling Algorithms for Ground & Aerial Robots: Precision Agriculture
  - Multi Target Tracking with Teams of Aerial Robots
  - Energy-optimal Trajectory Planning

Environment Coverage with Cameras

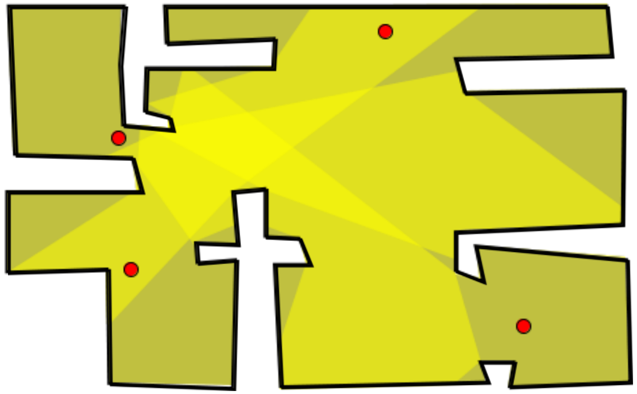
- Applications
  - *security, behavior analysis, trail cameras, forest-fire detection, etc.*
- Cameras are ubiquitous in robotics
- Understanding limitations of visibility is important

Art Gallery Problem



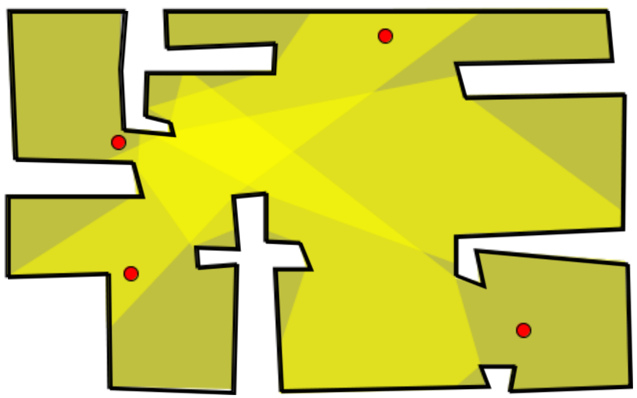
What is the minimum number of "guards" (omnidirectional cameras) sufficient to see every point in an environment?

Art Gallery Problem



What is the minimum number of "guards" (omnidirectional cameras) sufficient to see every point in an environment?

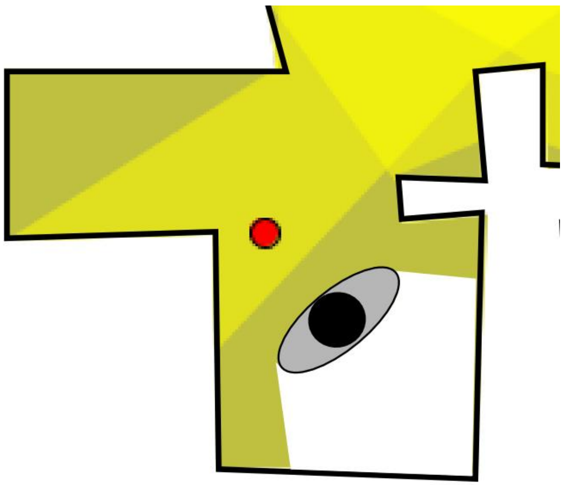
Art Gallery Problem



What is the minimum number of "guards" (omnidirectional cameras) sufficient to see every point in an environment?

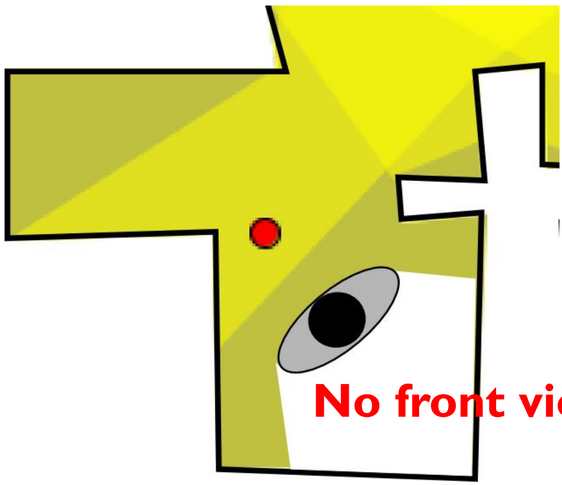
[Chvátal, 1975] $\lfloor n/3 \rfloor$ guards are sometimes necessary and always sufficient to see every point in an n-sided 2D polygonal environment.

Art Gallery Problem with Target Orientation



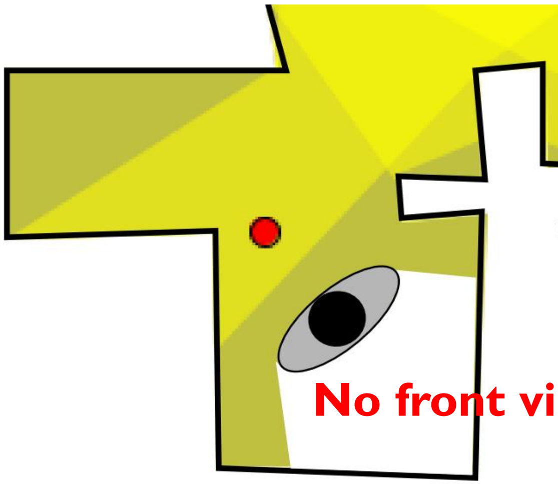
- Standard formulation does not handle self-occlusions

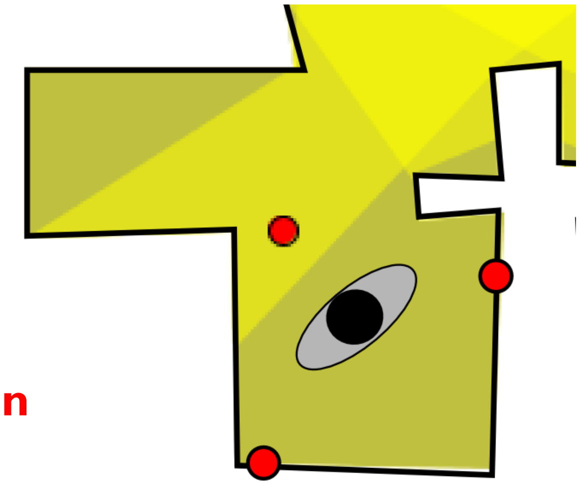
Art Gallery Problem with Target Orientation



- Standard formulation does not handle self-occlusions
- Many applications need a "good" view of the target
  - Surveillance, video conferencing, casinos!

Art Gallery Problem with Target Orientation

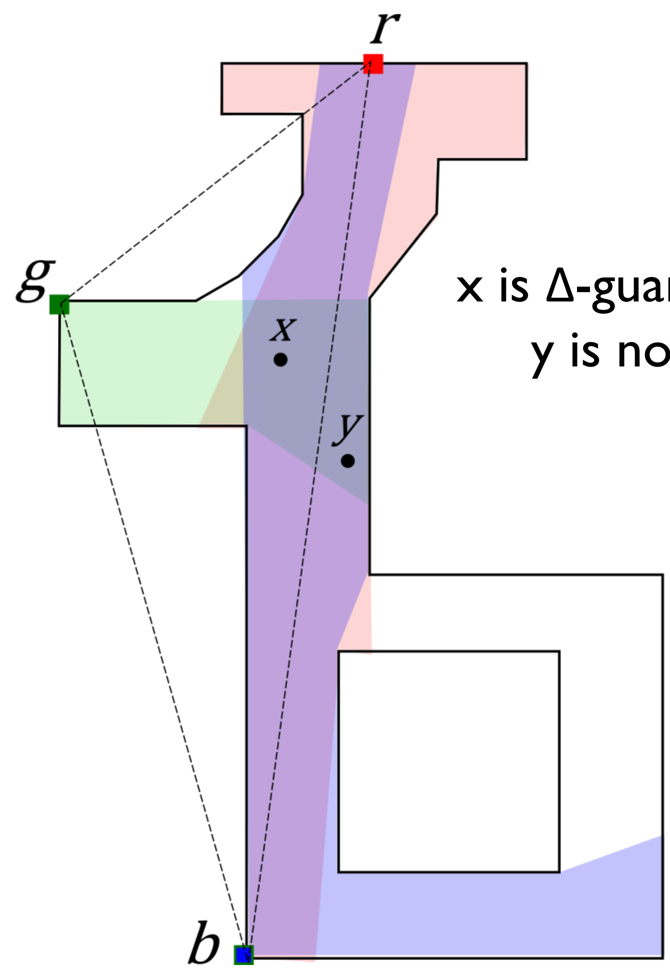




- Standard formulation does not handle self-occlusions
- Many applications need a “good” view of the target
  - Surveillance, video conferencing, casinos!

Δ-guarding Constraint

- Every point x is visible from a set of guards
- x lies in the convex hull of guards that see x
- Guards need not be visible from each other
- Introduced by [Smith & Evans, 2003]



x is Δ-guarded
y is not

Coverage with Orientation

- If all points in the polygon are Δ-guarded, then the perimeter of any convex object is completely visible.

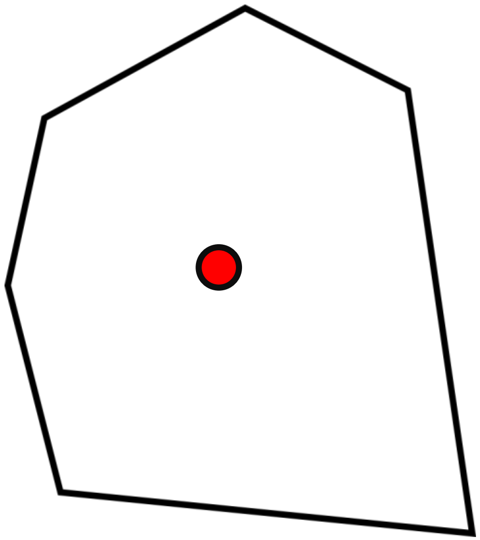
Coverage with Orientation

- If all points in the polygon are Δ-guarded, then the perimeter of any convex object is completely visible.

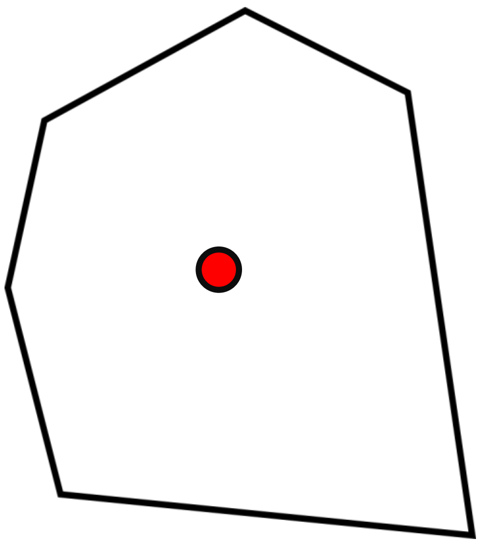
<u>Our Contributions</u>

1. What is the minimum number of guards required to Δ-guard a polygon?
2. Algorithms to place guards for Δ-guarding polygons.

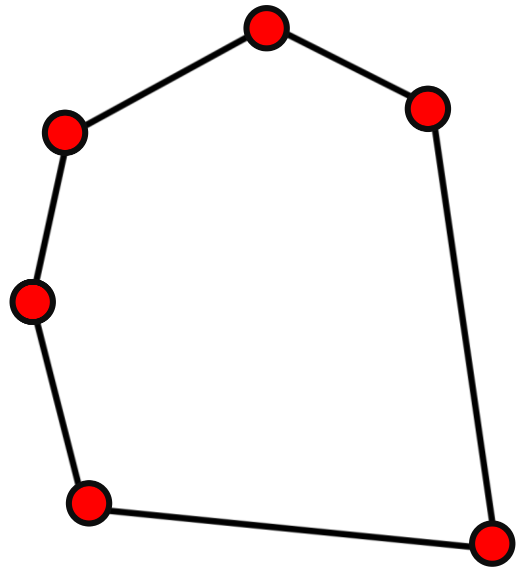
Without Δ-guarding



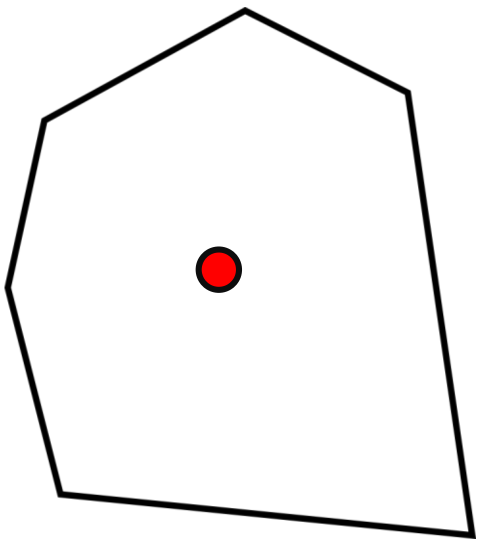
Without Δ-guarding



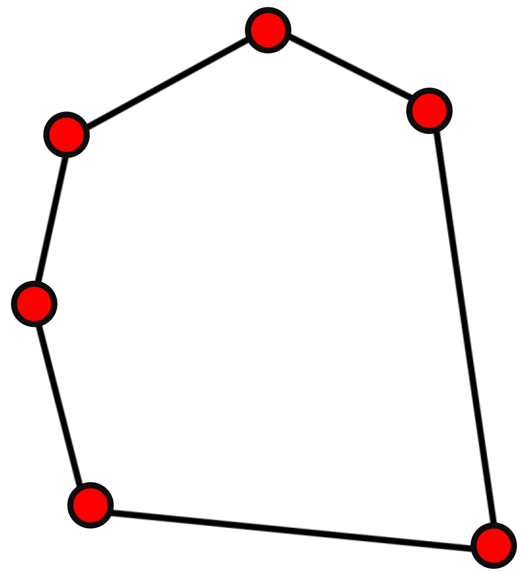
With Δ-guarding



Without Δ-guarding

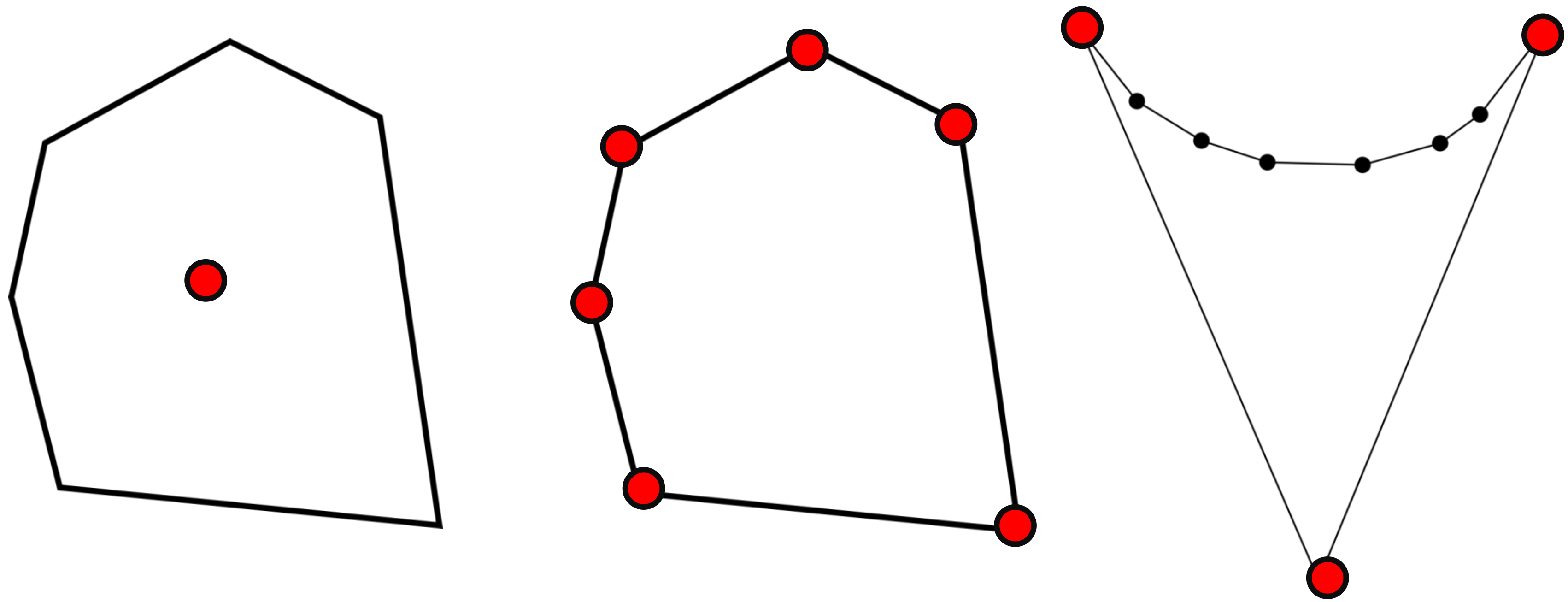


With Δ-guarding



Lemma 1: *There exists a guard on every convex vertex in any valid solution for Δ-guarding* ***any polygon****.*

Without Δ-guarding **With Δ-guarding** **Only 3 convex vertices**



Lemma 1: *There exists a guard on every convex vertex in any valid solution for Δ-guarding* ***any polygon****.*

Lower Bound for Δ-guarding

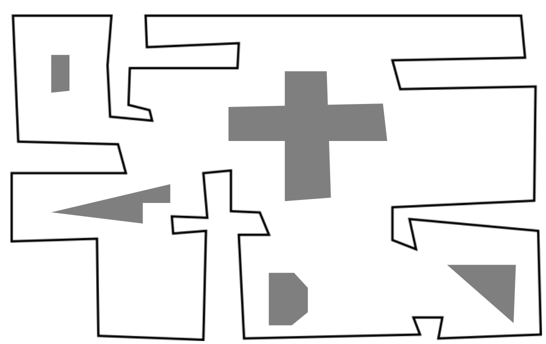
$\Omega(\sqrt{n})$ guards are always necessary for Δ-guarding any n-sided polygon (with or without holes).

Tokekar and Isler. **Polygon Guarding with Orientation**. ICRA 2014.

Lower Bound for Δ-guarding

$\Omega(\sqrt{n})$ guards are always necessary for Δ-guarding any n-sided polygon (with or without holes).

– Any n-sided polygon needs at least $\Omega(\sqrt{n})$ guards for Δ-guarding



Tokekar and Isler. **Polygon Guarding with Orientation.** ICRA 2014.

Proof Overview

- n_c number of convex vertices
- n_r number of reflex vertices
- $n = n_c + n_r$ total number of vertices

- Case 1: $n_c \geq n/4$

- Case 2: $n_c < n/4$

Proof Overview

- n_c number of convex vertices
- n_r number of reflex vertices
- $n = n_c + n_r$ total number of vertices

- Case 1: $n_c \geq n/4$

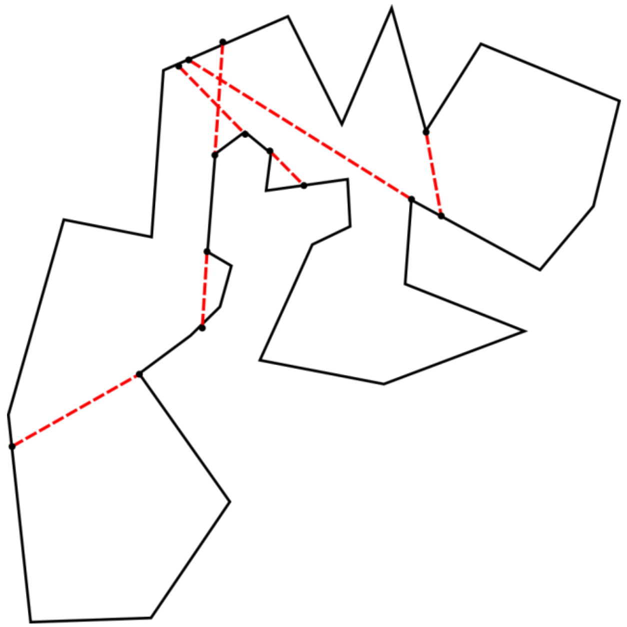
  Guard on every convex vertex

  $|G| \geq n_c \geq n/4 = \Omega(\sqrt{n})$

- Case 2: $n_c < n/4$

Case (2). $n_c < n/4$

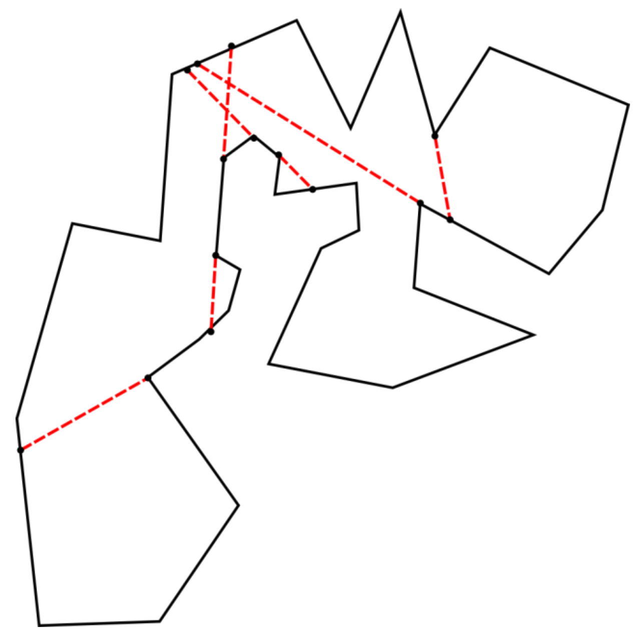
Edge extensions: Segments obtained by extending an edge on either side till they hit the boundary.



- No extension at convex vertices
- Each edge introduces up to two edge extensions

Case (2). $n_c < n/4$

Edge extensions: Segments obtained by extending an edge on either side till they hit the boundary.



- No extension at convex vertices
- Each edge introduces up to two edge extensions

Lemma 2: *There exists a guard on every edge extension in any valid solution for Δ-guarding a polygon.*

Case (2). $n_c < n/4$

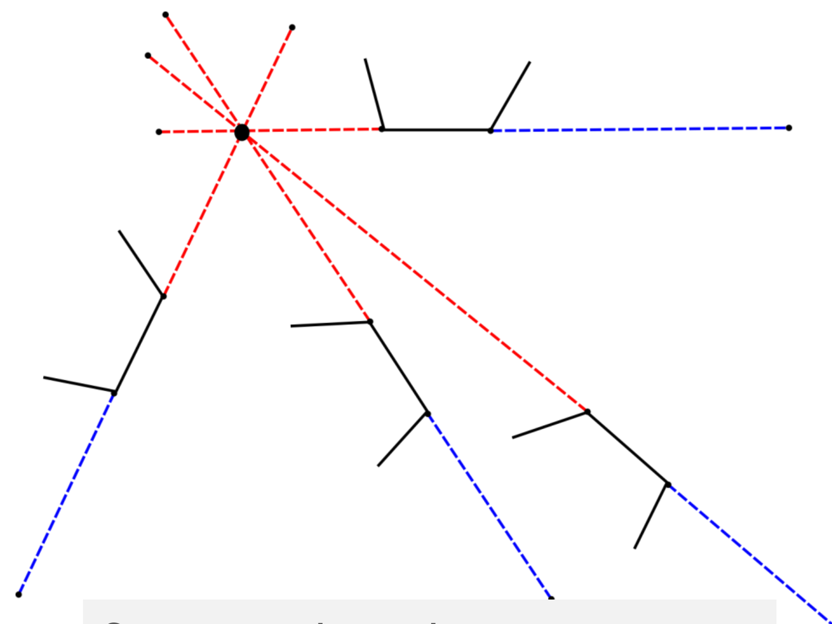
- **m:** number of edges incident with two reflex vertices
- **2m:** corresponding edge extensions

Is $|G| \geq 2m$?

Case (2). $n_c < n/4$

- **m:** number of edges incident with two reflex vertices
- **2m:** corresponding edge extensions

Is |G| ≥ 2m? No.

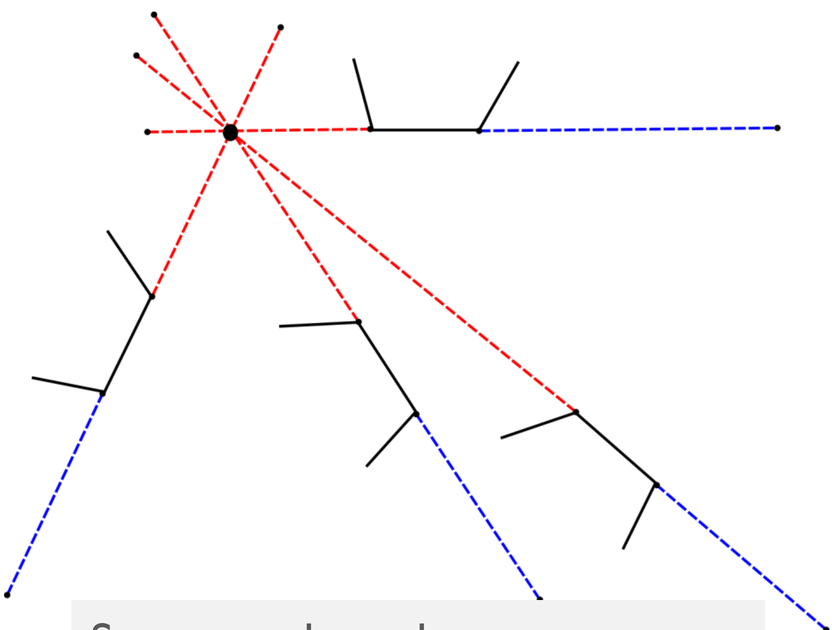


Same guard may be present on multiple extensions if they intersect at a point.

Case (2). $n_c < n/4$

- **m:** number of edges incident with two reflex vertices
- **2m:** corresponding edge extensions
- **k**: max. number of such extensions that intersect at a point
- Any guard covers at most **k** extensions
- Therefore, **|G| ≥ 2m/k**

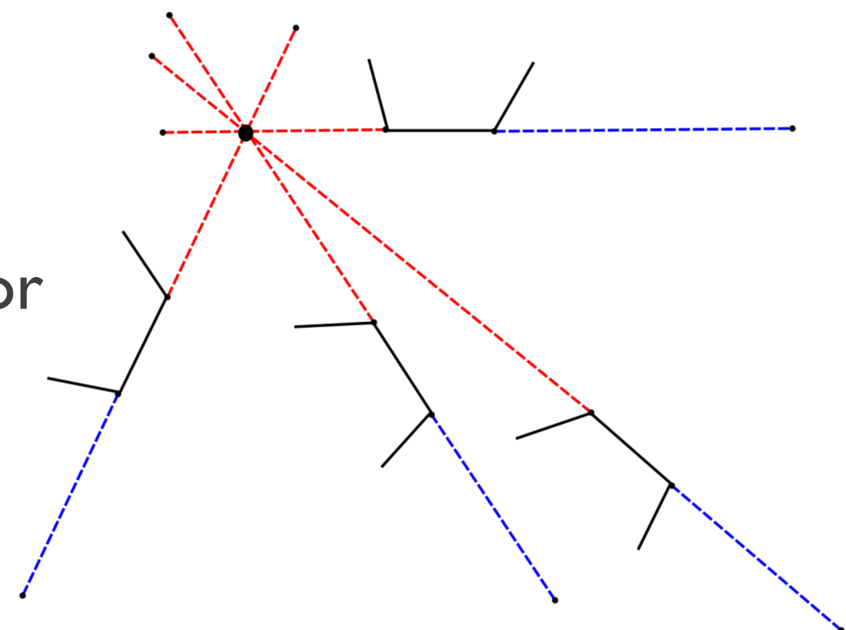
Is |G| ≥ 2m? No.



Same guard may be present on multiple extensions if they intersect at a point.

Case (2). $n_c < n/4$

- **|G| ≥ 2m/k**
- If red extensions intersect, blue cannot
  - Therefore, separate guards for all blue extensions
- **|G| ≥ k**
- Combining, **$|G|^2$ ≥ 2m**
- m: reflex-reflex edges
  - **m ≥ n-2nc ≥ n/2**



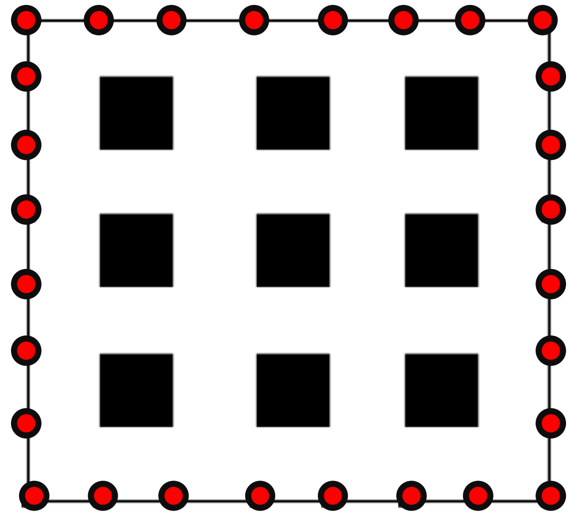
Each edge corresponding to these **k** extensions (red) also has another extension (blue)

Recap

- Case 1: $n_c \geq n/4$
  - Guard on every convex vertex
- Case 2: $n_c < n/4$
  - Guard on every edge extension
- $\Omega(\sqrt{n})$ are always necessary

Recap

- Case 1: $n_c \geq n/4$
  - Guard on every convex vertex
- Case 2: $n_c < n/4$
  - Guard on every edge extension
- $\Omega(\sqrt{n})$ are always necessary
  - and sometimes sufficient (for polygons with holes)



n= $4k^2$ + 4 vertices

8k + 4 guards ●

Guard Placement

If optimal algorithm uses k guards, c-approx. algorithm uses at most ck guards.

- Placement algorithms are difficult
  - only a few problems have constant-factor approximation algorithms

Guard Placement

If optimal algorithm uses k guards, c-approx. algorithm uses at most ck guards.

- Placement algorithms are difficult
  - only a few problems have constant-factor approximation algorithms
- $O(\log |G^*|)$ approximation with greedy algorithm (guards restricted to vertices)

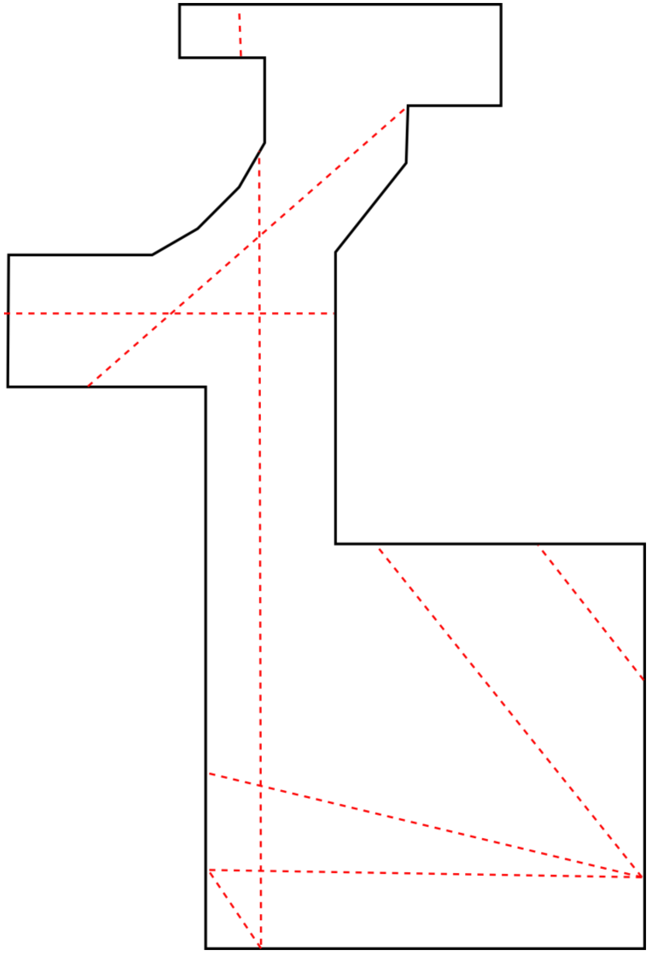
Guard Placement

If optimal algorithm uses k guards, c-approx. algorithm uses at most ck guards.

- Placement algorithms are difficult
  - only a few problems have constant-factor approximation algorithms
- $O(\log |G^*|)$ approximation with greedy algorithm (guards restricted to vertices)
- $\sqrt{n}$ is too high
  - Guarding all corners and edges may not be necessary
  - Guard only the region of interest

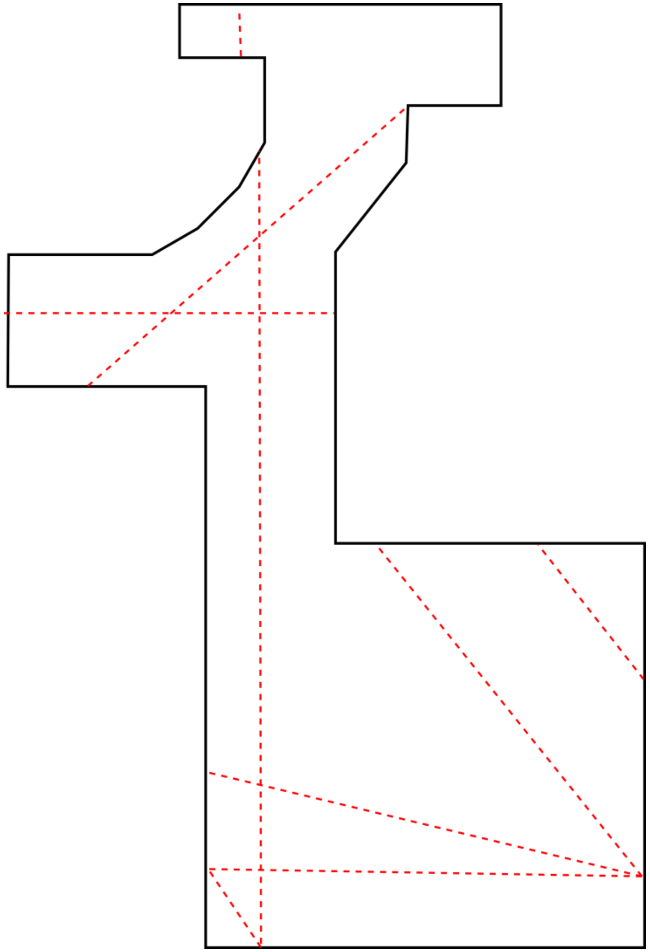
Guarding Paths

- **Chord**: line segment between two visible points on the boundary
  - e.g. Target paths

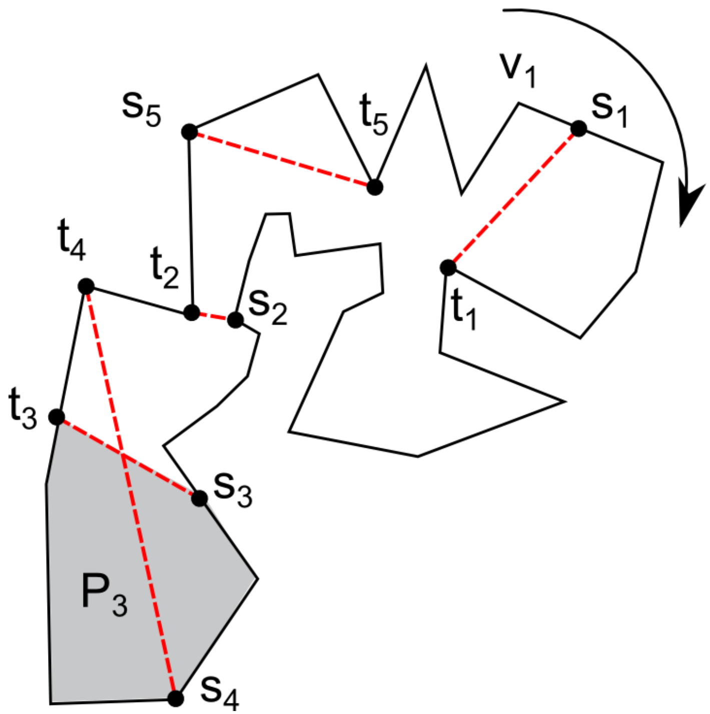


Guarding Paths

- **Chord**: line segment between two visible points on the boundary
  - e.g. Target paths
- **Given:** Set of chords in simply-connected polygon
- **Objective:** Δ-guard at least one point on each chord
- **Result:** 12-approximation algorithm.

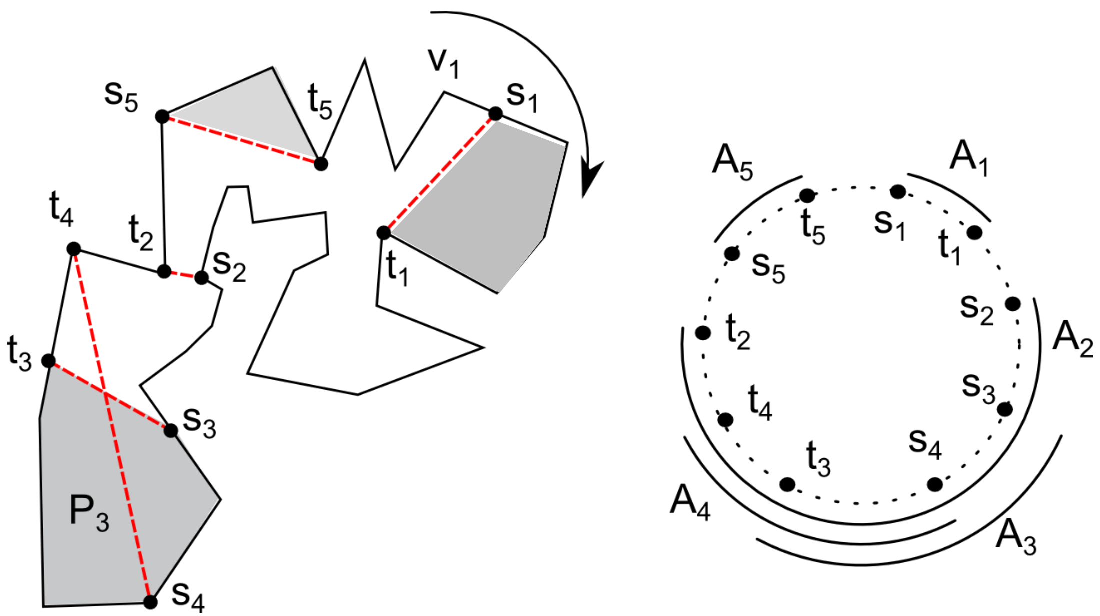


Main Idea (1/2)



- Each chord partitions polygon into two subpolygons
- If chord s_3t_3 is Δ-guarded, then there exists a guard used in Δ-guarding s_3t_3 in **both** subpolygons
  - **There exists a guard in P_3**

Main Idea (2/2)



- Find maximum set of chords whose subpolygons are disjoint
- |G| ≥ |maximum number of disjoint subpolygons|
- Place four guards per disjoint subpolygon to guard a subset of chords

Δ-guarding Recap

- All n-sided polygons need at least $\Omega(\sqrt{n})$ guards
- $O(\log |G^*|)$ approximation for Δ-guarding any polygon with holes (guards restricted to vertices)
- Constant-factor approximation algorithm for Δ-guarding a set of chords

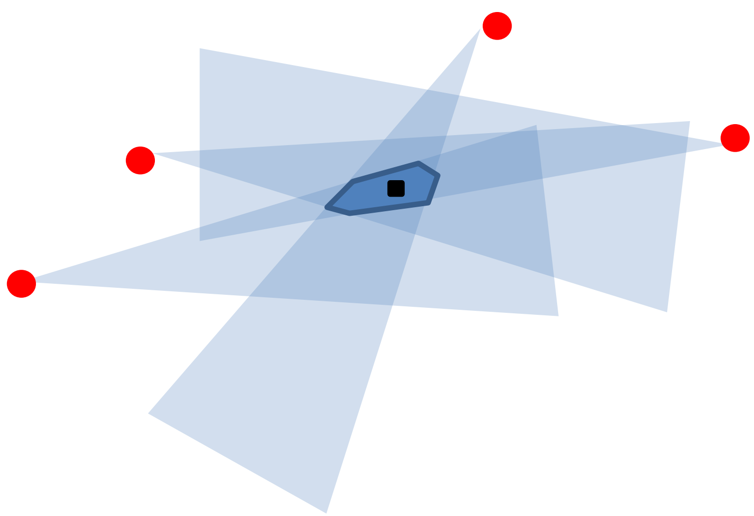
- **Future Work:**
  - Tight lower bound for polygons without holes
  - Other types of regions of interest

Gap between Theory and Practice

- Practical constraints
  - sensor measurements are noisy
  - may not be placed precisely
  - modeling realistic environments

Gap between Theory and Practice

- Practical constraints
  - sensor measurements are noisy
  - may not be placed precisely
  - modeling realistic environments



- Unknown but bounded noise in measurements
- Trade-off between number of sensors and uncertainty
- Adversarial guarantees under worst-case measurements

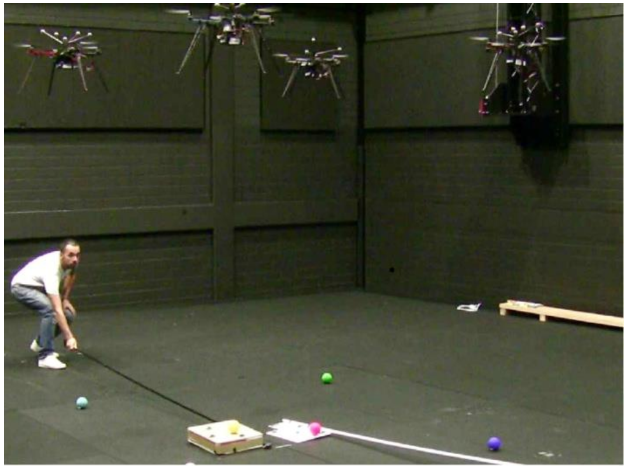
Tokekar and Isler. **Sensor Placement and Selection for Bearing Sensors with Bounded Uncertainty**. ICRA, 2013 & submitted to T-ASE.

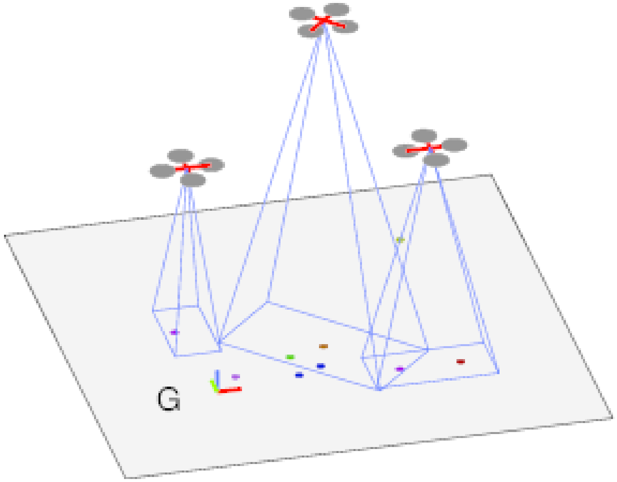
Thesis Contributions

- **Placement for Stationary Sensors**
  - Visibility-based Coverage with Orientation
  - Target Localization with Bearing Sensors
- **Motion Planning for Mobile Sensors**
  - Multi Target Tracking with Teams of Aerial Robots
  - Coverage and Active Localization with Robotic Boats: Carp Monitoring
  - Sampling Algorithms for Ground & Aerial Robots: Precision Agriculture
  - Energy-optimal Trajectory Planning

Visual Tracking with Robots

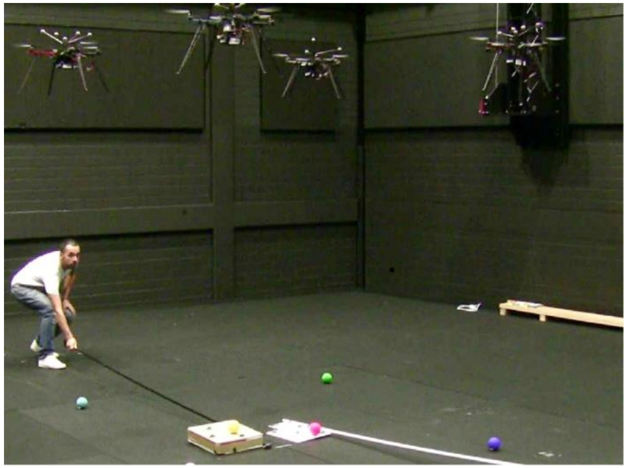
- k aerial robots carrying cameras
- n targets moving on the ground
- How should the robots move in order to track all targets?

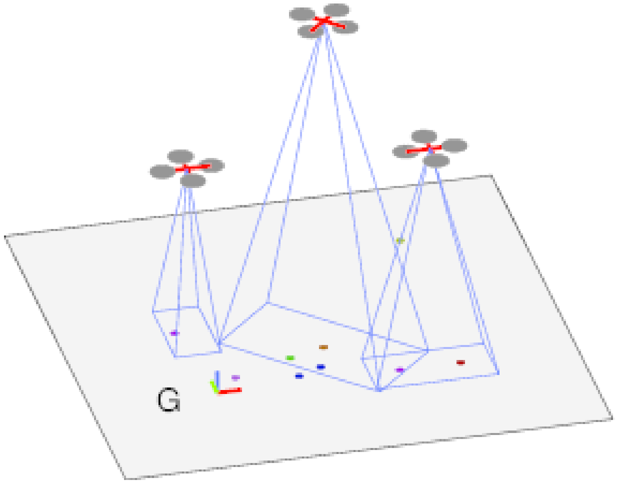




Visual Tracking with Robots

- k aerial robots carrying cameras
- n targets moving on the ground
- How should the robots move in order to track all targets?
- Camera footprint increases as altitude increases
- Area per pixel increases (i.e., resolution/quality decreases) as altitude increases





Tracking Quality vs. Number of Targets Tracked

- k robots cannot always track more than k targets while maintaining the optimal tracking quality at all times, even if

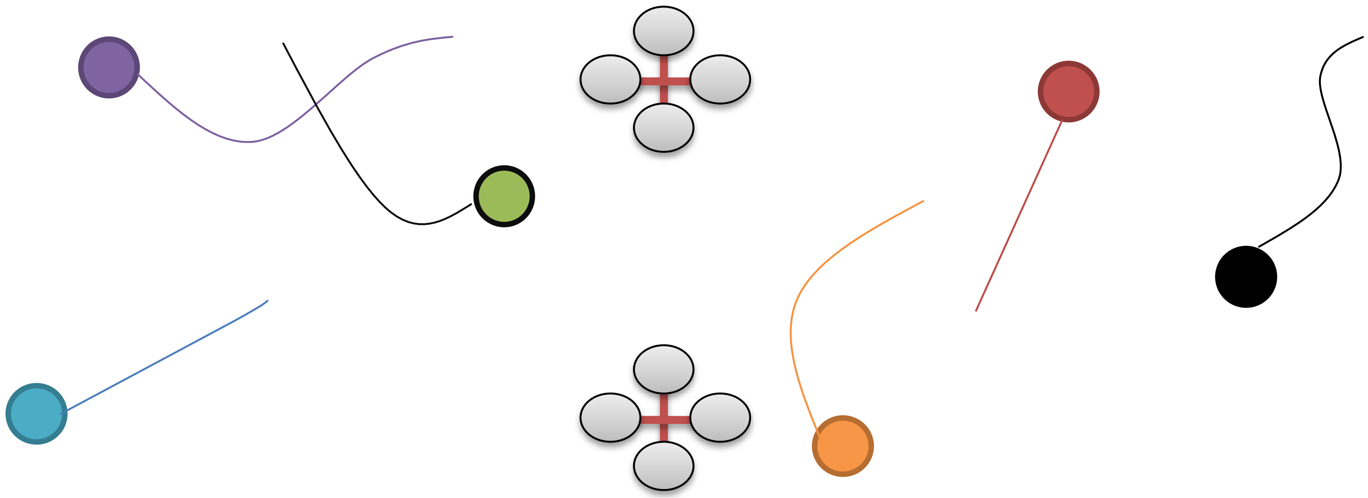
Tracking Quality vs. Number of Targets Tracked

- k robots cannot always track more than k targets while maintaining the optimal tracking quality at all times, even if
  - the robots are faster than the targets (but still have bounded speeds)
  - targets move on one line with constant speed
  - only an approximation of the optimal quality of tracking is to be maintained

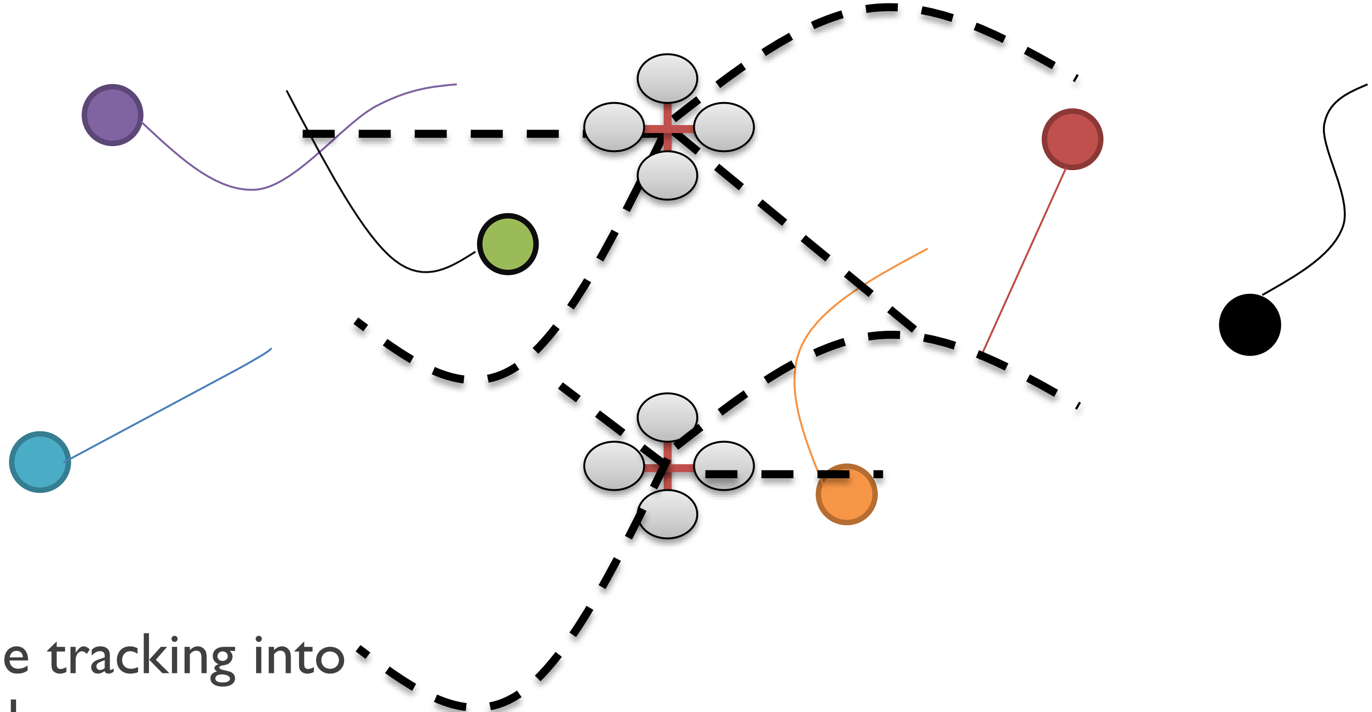
Tracking Quality vs. Number of Targets Tracked

- k robots cannot always track more than k targets while maintaining the optimal tracking quality at all times, even if
  - the robots are faster than the targets (but still have bounded speeds)
  - targets move on one line with constant speed
  - only an approximation of the optimal quality of tracking is to be maintained

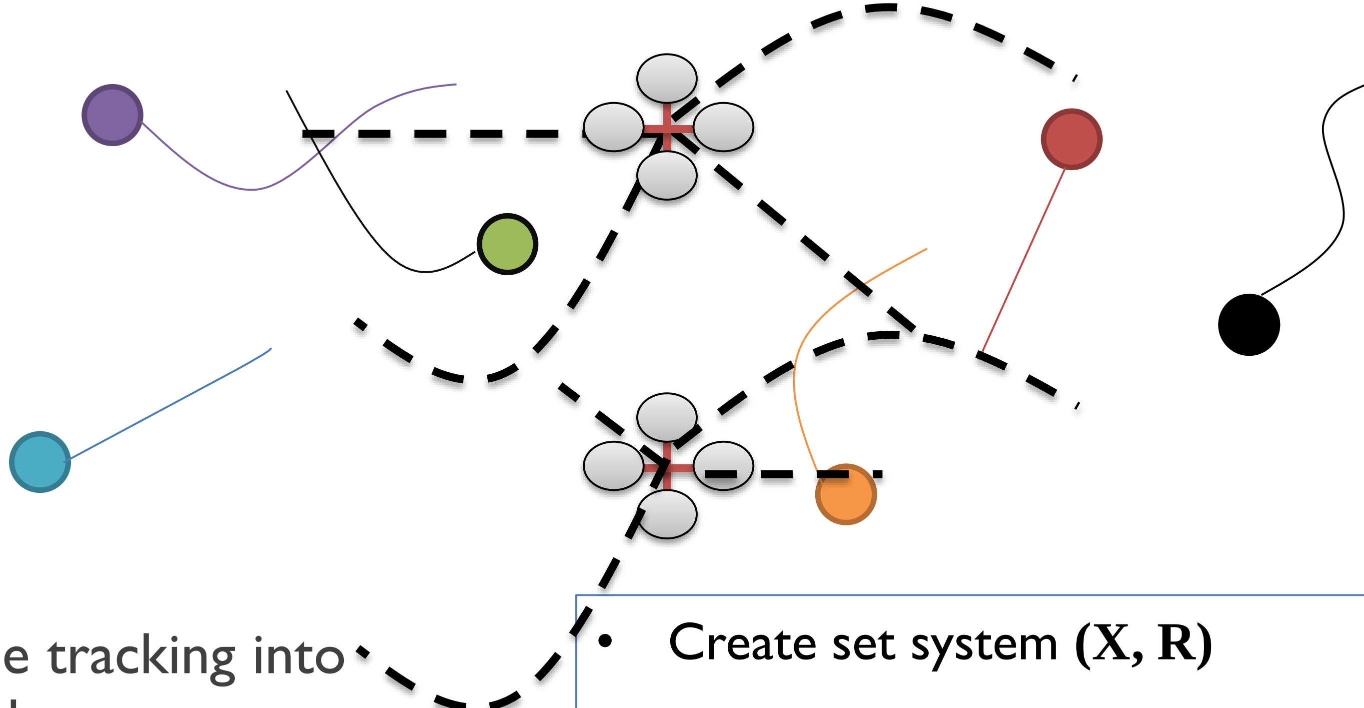
- Instead, we formulate two versions
  - Maximize the number of targets tracked
  - Maximize the total quality of tracking



- Divide tracking into rounds
- Predict target motion for the next round
- Generate m candidate trajectories per robot
- Assign one trajectory per robot

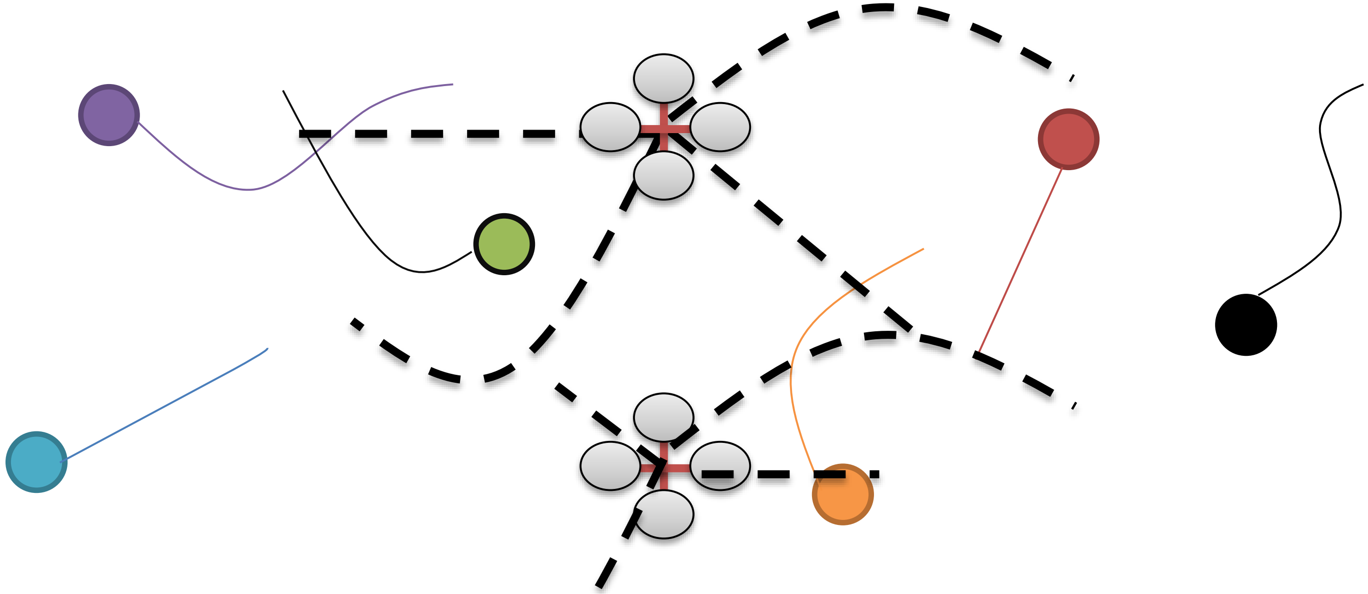


- Divide tracking into rounds
- Predict target motion for the next round
- Generate m candidate trajectories per robot
- Assign one trajectory per robot



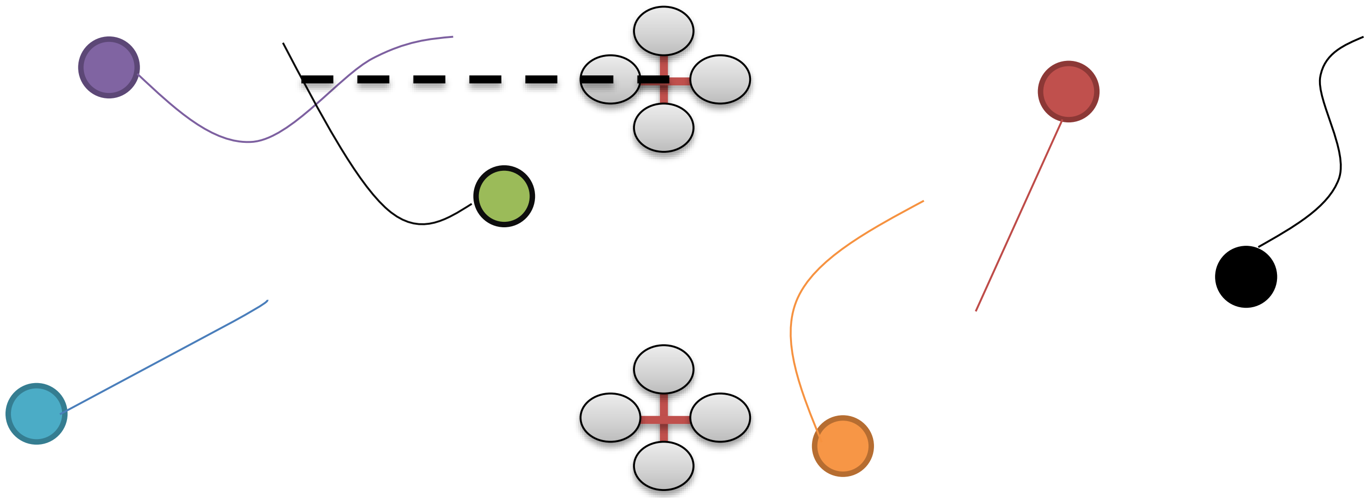
- Divide tracking into rounds
- Predict target motion for the next round
- Generate m candidate trajectories per robot
- Assign one trajectory per robot

- Create set system **(X, R)**



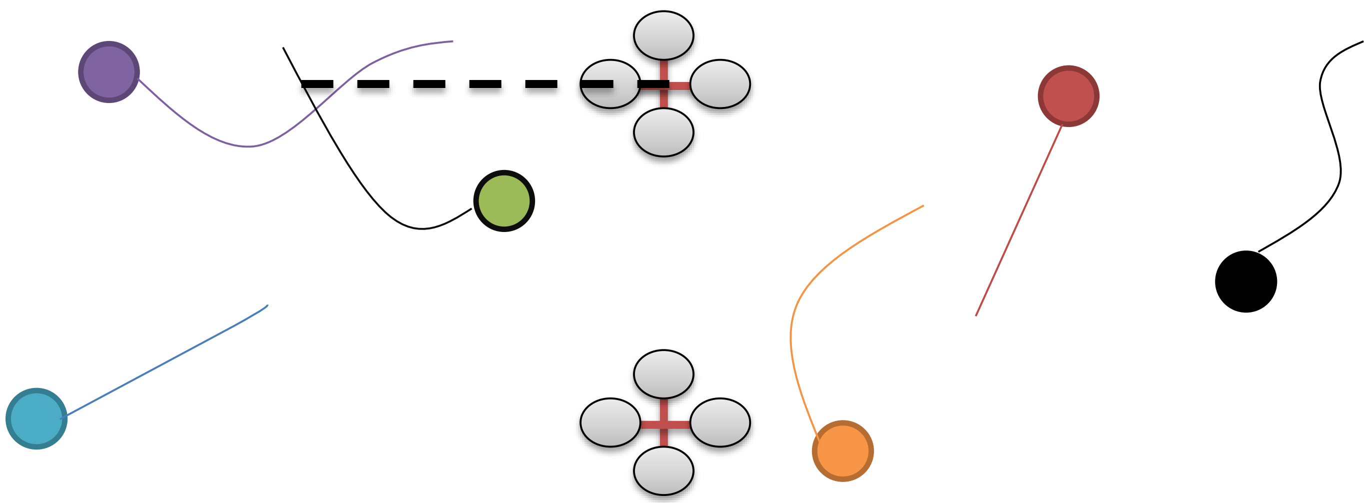
- Divide tracking into rounds
- Predict target motion for the next round
- Generate m candidate trajectories per robot
- Assign one trajectory per robot

- Create set system **(X, R)**
- **X** = { , , , , , }



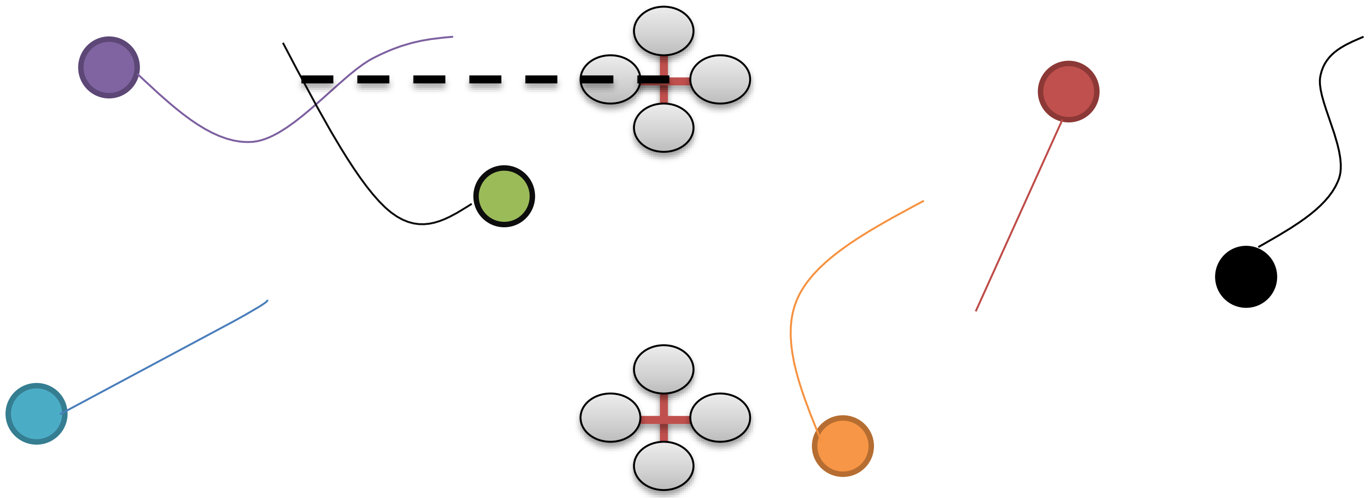
- Divide tracking into rounds
- Predict target motion for the next round
- Generate m candidate trajectories per robot
- Assign one trajectory per robot

- Create set system **(X, R)**
- **X** = { , , , , , }
- $R_{i,j}$: targets tracked by i^{th} robot with j^{th} trajectory
  - $R_{1,1}$ = { , }



- Divide tracking into rounds
- Predict target motion for the next round
- Generate m candidate trajectories per robot
- Assign one trajectory per robot

- Create set system **(X, R)**
- **X** = { , , , , , }
- $R_{i,j}$: targets tracked by i^{th} robot with j^{th} trajectory
  - $R_{1,1} = \{ \, , \, \}$
- **R** = {$R_{1,1}$, … , $R_{1,m}$, …… , $R_{k,1}$, … , $R_{k,m}$}



- Divide tracking into rounds
- Predict target motion for the next round
- Generate m candidate trajectories per robot
- Assign one trajectory per robot

- Create set system **(X, R)**
- $\mathbf{X} = \{\ ,\ ,\ ,\ ,\ ,\ \}$
- $R_{i,j}$: targets tracked by i^{th} robot with j^{th} trajectory
  - $R_{1,1} = \{\ ,\ \}$
- $\mathbf{R} = \{R_{1,1}, \ldots, R_{1,m}, \ldots\ldots, R_{k,1}, \ldots, R_{k,m}\}$
- Choose k sets in R, one per group, to maximize the weight of their union

Max-Cover with Group Constraints

- Generalization of max-cover problem
  - Choose any k sets in $\mathbf{R}$ (no constraint of choosing one per group)

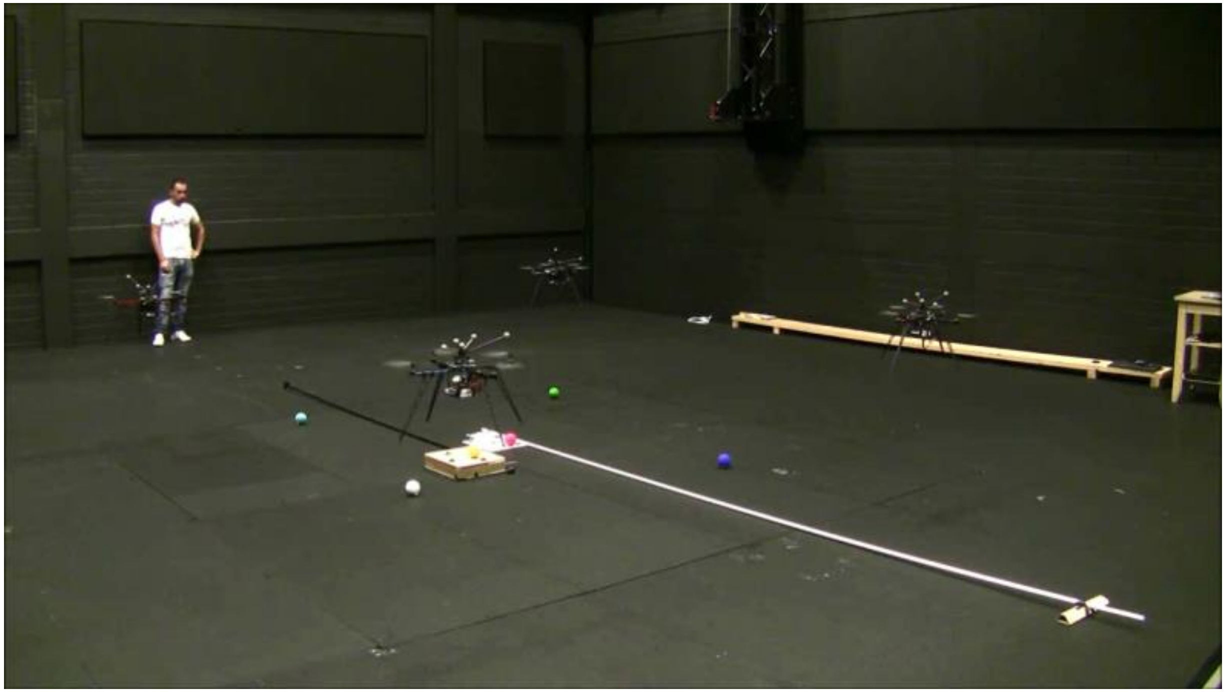
Max-Cover with Group Constraints

- Generalization of max-cover problem
  - Choose any k sets in $\mathbf{R}$ (no constraint of choosing one per group)
- Greedy yields a 2-approximation with for unweighted max-cover with group constraints [Chekuri & Kumar, '04]
  - Maximize the total number of targets tracked

Max-Cover with Group Constraints

- Generalization of max-cover problem
  - Choose any k sets in $\mathbf{R}$ (no constraint of choosing one per group)
- Greedy yields a 2-approximation with for unweighted max-cover with group constraints [Chekuri & Kumar, '04]
  - Maximize the total number of targets tracked
- We formulate the problem of maximizing the total quality of tracking as the weighted version
  - Greedy yields 2-approximation in this case as well

Preliminary Experiments



- **Future Work**
  - The infeasibility result is valid only for unbounded environments
  - Stronger guarantees for special classes of target motion models
  - Inter-robot communication

Thesis Contributions

- **Placement for Stationary Sensors**
  - Visibility-based Coverage with Orientation
  - Target Localization with Bearing Sensors
- **Motion Planning for Mobile Sensors**
  - Multi Target Tracking with Teams of Aerial Robots
  - Coverage and Active Localization with Robotic Boats: Carp Monitoring
  - Sampling Algorithms for Ground & Aerial Robots: Precision Agriculture
  - Energy-optimal Trajectory Planning

Bottleneck: On-board Battery

< 20 mins

< 30 mins

~ 1.5 hours

~ 2 hours

- Optimize low-level motion to reduce energy consumption
- Incorporate energy constraints in high-level planning
- Design energy efficient systems
- Harvest additional energy

Energy-optimal Path and Velocity Profiles

- Dubins' car kinematics
  - DC motor for translation

$$E = \int_0^{t_f} \left[c_1 a^2(t) + c_2 v^2(t) + c_3 v(t) + c_4 \right] \mathrm{d}t$$

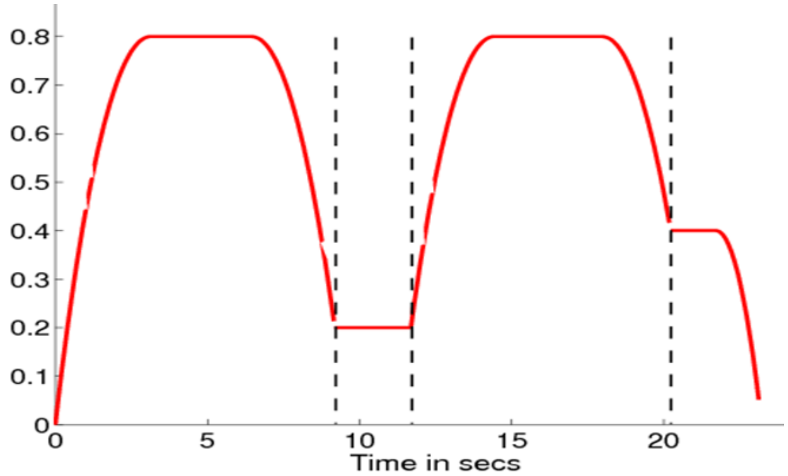
Acceleration **Velocity** **Constant (Time)**

Tokekar, Karnad and Isler. **Energy-Optimal Trajectory Planning for Car-Like Robots.** Autonomous Robots, 2014.

Energy-optimal Path and Velocity Profiles

- Dubins' car kinematics
  - DC motor for translation

Closed-form Energy-Optimal Velocity using Optimal Control

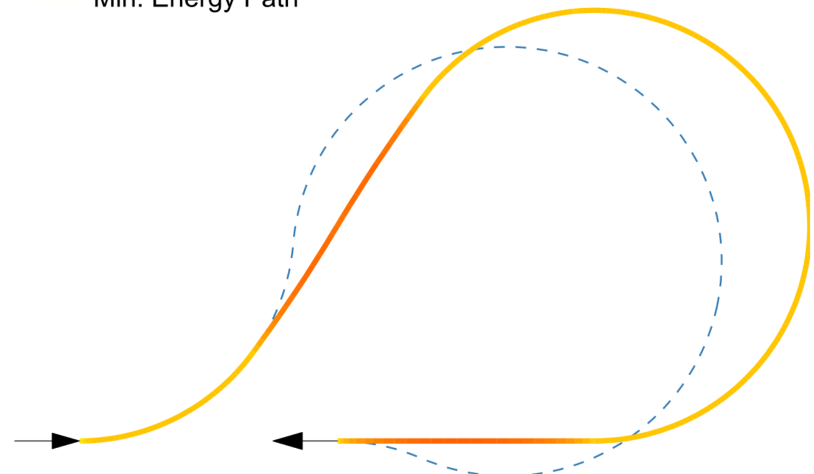


$$E = \int_0^{t_f} \left[\underbrace{c_1 a^2(t)}_{\text{Acceleration}} + \underbrace{c_2 v^2(t) + c_3 v(t)}_{\text{Velocity}} + \underbrace{c_4}_{\text{Constant (Time)}}\right] \mathrm{d}t$$

Tokekar, Karnad and Isler. **Energy-Optimal Trajectory Planning for Car-Like Robots.** Autonomous Robots, 2014.

Energy-optimal Path and Velocity Profiles

- Dubins' car kinematics
  - DC motor for translation

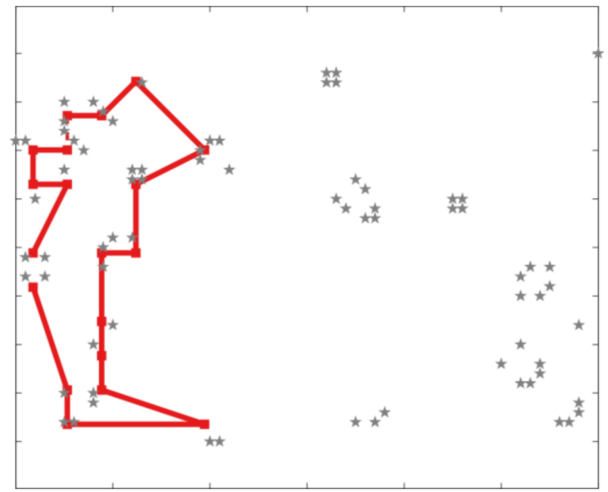


$$E = \int_0^{t_f} \Big[\underbrace{c_1 a^2(t)}_{\textbf{Acceleration}} + \underbrace{c_2 v^2(t) + c_3 v(t)}_{\textbf{Velocity}} \underbrace{+ c_4}_{\textbf{Constant (Time)}} \Big] \, \mathrm{d}t$$

Tokekar, Karnad and Isler. **Energy-Optimal Trajectory Planning for Car-Like Robots.** Autonomous Robots, 2014.

Coverage with an Aerial Robot

- Obtain aerial images at given points
- **UAVs have limited battery life**
- May not be able to visit all points
- Visit most number of points: Orienteering

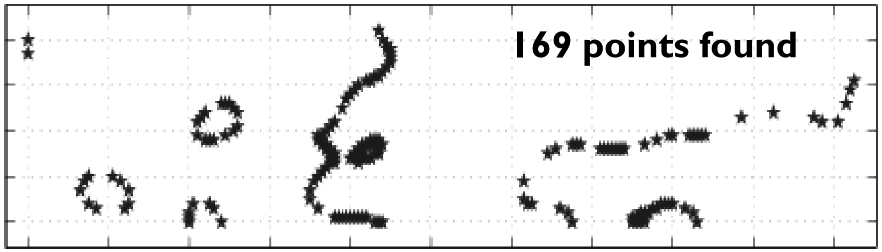


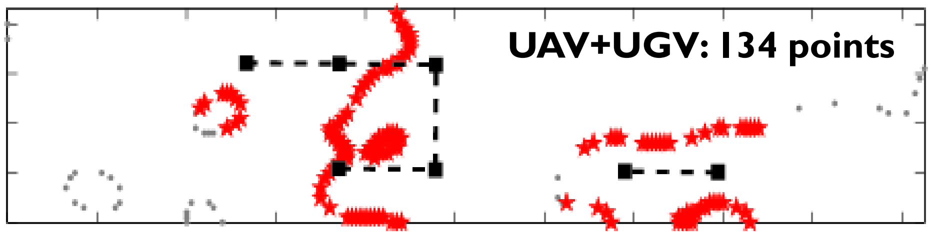
Coverage with an Aerial Robot

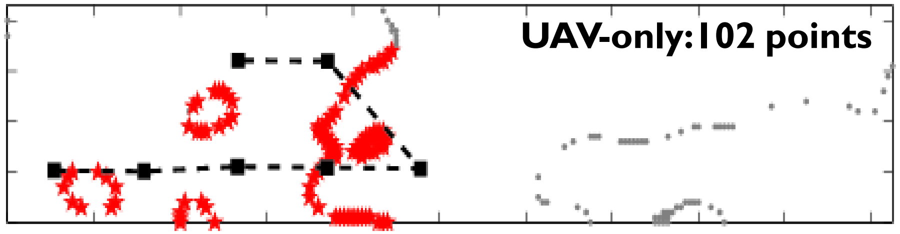
- Obtain aerial images at given points
- **UAVs have limited battery life**
- May not be able to visit all points
- Visit most number of points: Orienteering

- **UAV+UGV System**
  - UAV can land on UGV
  - UGV carries UAV between deployment locations.
- **How to plan such paths?**
  - We show how to formulate the problem as an orienteering instance.

Plots from Precision Agriculture Experiments







Thesis Contributions

- **Placement for Stationary Sensors**
  - Visibility-based Coverage with Orientation
    [ICRA '14]
  - Target Localization with Bearing Sensors
    [ICRA '13] [Submitted to T-ASE]

- Devise algorithms with theoretical performance guarantees
- Prototype system design
- Evaluate algorithms through field experiments

- **Motion Planning for Mobile Sensors**
  - Coverage and Active Localization with Robotic Boats: Carp Monitoring
    [ICRA '10, IROS '11] [JFR '10, R&A Magazine '13]
  - Sampling Algorithms for Ground & Aerial Robots: Precision Agriculture
    [IROS '13] [Submitted to T-RO]
  - Multi Target Tracking with Teams of Aerial Robots
    [IROS '14]
  - Energy-optimal Trajectory Planning
    [ICRA '11] [AURO '14]

Future Research Directions

- Realistic environments
  - Instead of arbitrary polygons, consider only "realistic" polygons
    - How to define "realistic"? fat polygons, ε-good polygons, data-driven approaches
  - Uncertainty in specifying the polygon
    - "robust" guarding
- Realistic motion models for the robots
  - We modeled robots as points which allows us to obtain stronger guarantees but possibly with low fidelity
  - Hierarchy of results: e.g., coverage with
    - point model
    - steering constraints (Dubins', differential drive)
    - dynamics (friction, drift, etc.)

Thanks!

- My adviser: Prof. Volkan Isler
- Thesis committee members
- Collaborators
- Computer Science Department
- National Science Foundation
- Family and friends
- Former and current members of the RSN Lab

Thank you!

tokekar@cs.umn.edu