Visibility-based multiagent deployment in orthogonal environments

Francesco Bullo

Center for Control,
Dynamical Systems & Computation
University of California at Santa Barbara
http://motion.mee.ucsb.edu

American Control Conference, New York, July 12, 2007

Co-authors: Anurag Ganguli, Jorge Cortés

Outline

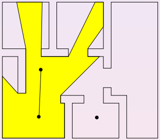
Outline

1. Robotic agents with visibility sensors
2. Deployment of multiple agents in orthogonal environments
3. Conclusions

Robotic agents with visibility sensors

- **Orthogonal polygon**
  Q: adjacent edges perpendicular to each other
- **Visibility**

  Visibility polygon
  $\mathcal{V}(p, Q) = \{q \in Q \mid q \text{ is visible from } p\}$



- **Robotic agent**
  First order dynamics: $p(t+1) = p(t) + u$
  Point robot with omnidirectional visibility sensing
  Line of sight communication: visibility graph

Outline

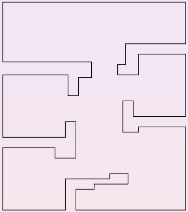
1. Robotic agents with visibility sensors
2. Deployment of multiple agents in orthogonal environments
3. Conclusions

Art Gallery Problem (Klee '73):

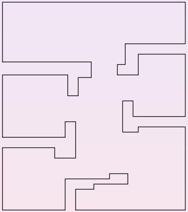
Imagine placing guards inside a nonconvex polygon with n vertices: how many guards are required and where should they be placed in order for each point in the polygon to be visible by at least one guard?

Art Gallery Problem (Klee '73):

Imagine placing guards inside a nonconvex polygon with n vertices: how many guards are required and where should they be placed in order for each point in the polygon to be visible by at least one guard?



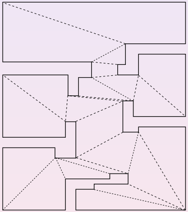
- Kahn et al '93
- $\lfloor \frac{n}{4} \rfloor$ sufficient and occasionally necessary



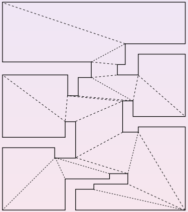
- Pinciu '03
- $\frac{n}{2} - 2$ sufficient and occasionally necessary

Art Gallery Problem (Klee '73):

Imagine placing guards inside a nonconvex polygon with n vertices: how many guards are required and where should they be placed in order for each point in the polygon to be visible by at least one guard?



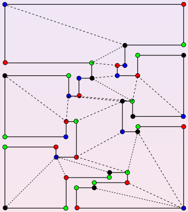
- Kahn et al '93
- $\lfloor \frac{n}{4} \rfloor$ sufficient and occasionally necessary

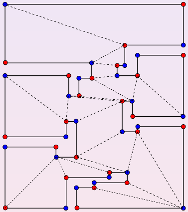


- Pinciu '03
- $\frac{n}{2} - 2$ sufficient and occasionally necessary

Art Gallery Problem (Klee '73):

Imagine placing guards inside a nonconvex polygon with n vertices: how many guards are required and where should they be placed in order for each point in the polygon to be visible by at least one guard?



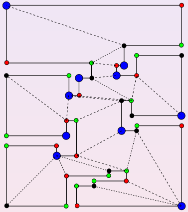


- Kahn et al '93
- $\lfloor \frac{n}{4} \rfloor$ sufficient and occasionally necessary

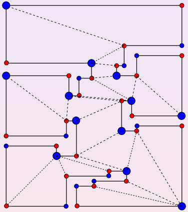
- Pinciu '03
- $\frac{n}{2} - 2$ sufficient and occasionally necessary

Art Gallery Problem (Klee '73):

Imagine placing guards inside a nonconvex polygon with n vertices: how many guards are required and where should they be placed in order for each point in the polygon to be visible by at least one guard?



- Kahn et al '93
- $\lfloor \frac{n}{4} \rfloor$ sufficient and occasionally necessary



- Pinciu '03:
- $\frac{n}{2} - 2$ sufficient and occasionally necessary

Robotic network model

- Communicate within line-of-sight and within bounded distance
- Each agent has a unique identifier i
- p_i denotes position; $p_i(t + \Delta t) = p_i(t) + u_i$, $\|u_i\| \leq 1$
- $\mathcal{M}_i$ denotes memory ("limited") contents

Deployment problems

Nonconvex deployment problem

Design a provably correct distributed algorithm:

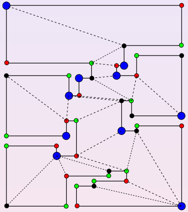
1. achieve complete visibility;
2. minimize the number of agents used

Nonconvex deployment problem with connectivity

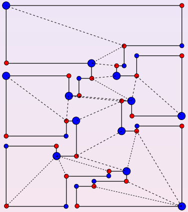
Design a provably correct distributed algorithm:

1. achieve complete visibility;
2. ensure that the visibility graph of final configuration is connected; and
3. minimize the number of agents used

Statement of results

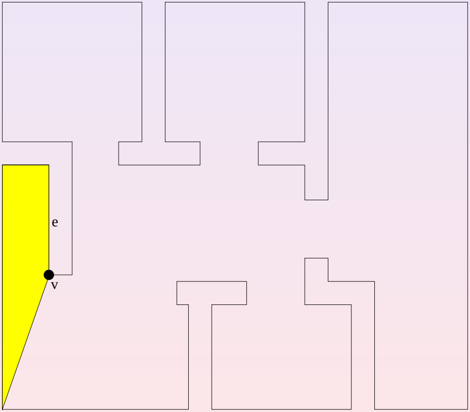


Starting from a single location, $\lfloor \frac{n}{4} \rfloor$ agents are always sufficient and occasionally necessary

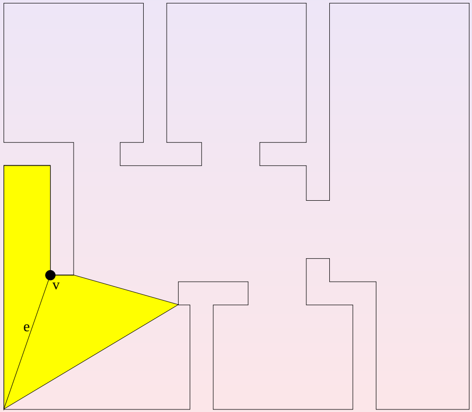


Starting from a single location, $\frac{n}{2} - 2$ are always sufficient and occasionally necessary

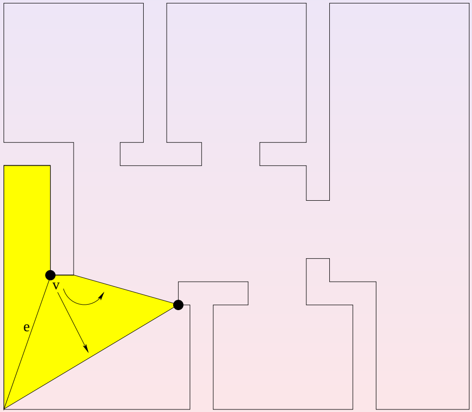
Incremental Partition Algorithm



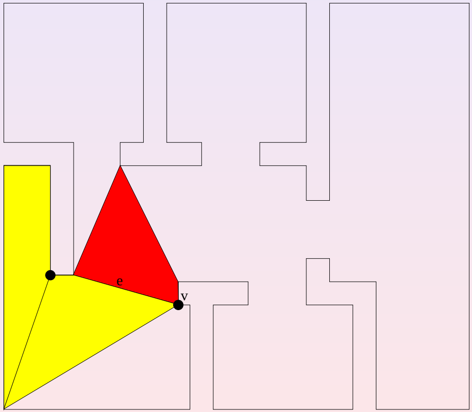
Incremental Partition Algorithm



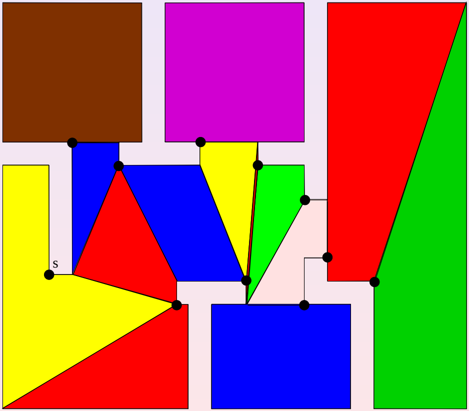
Incremental Partition Algorithm



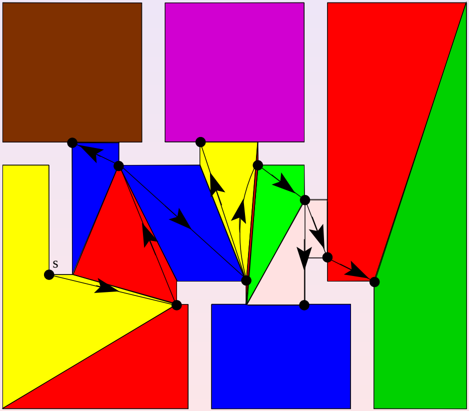
Incremental Partition Algorithm



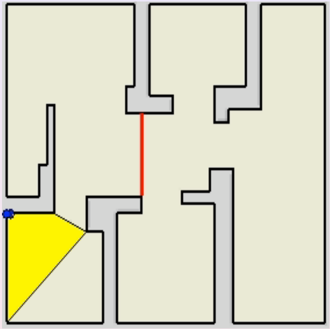
Incremental Partition Algorithm



Vertex-induced tree



Incremental algorithm for connected deployment

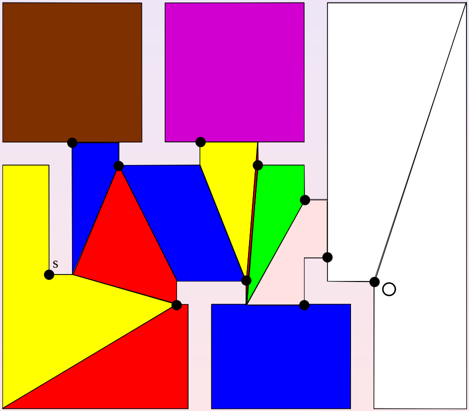


Robustness properties

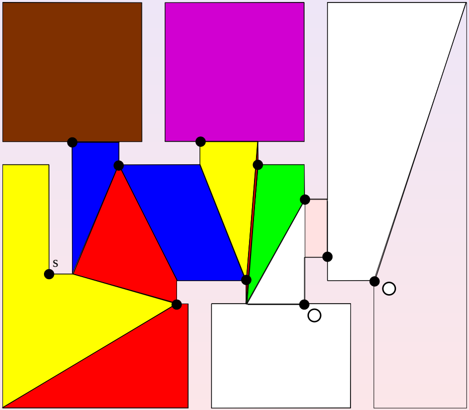
Robust to agent failures

Changing environments

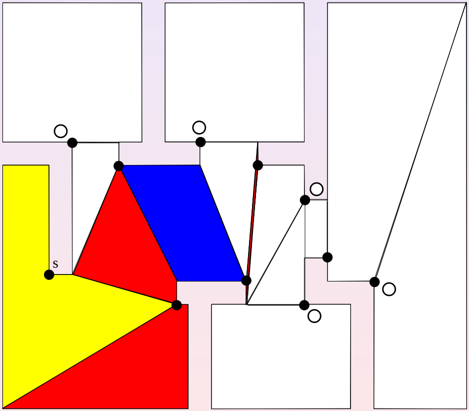
Sparse point set for deployment without connectivity



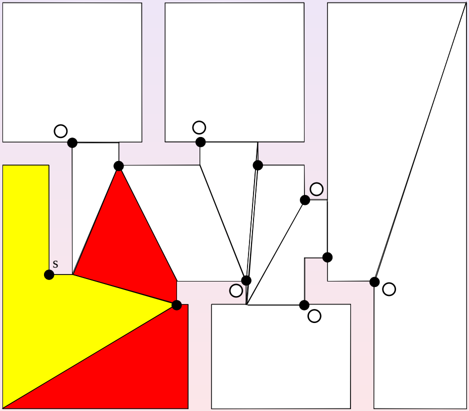
Sparse point set for deployment without connectivity



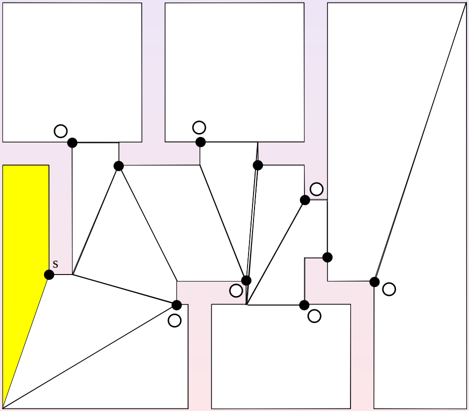
Sparse point set for deployment without connectivity



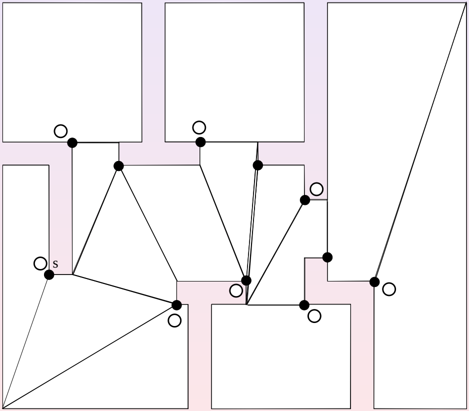
Sparse point set for deployment without connectivity



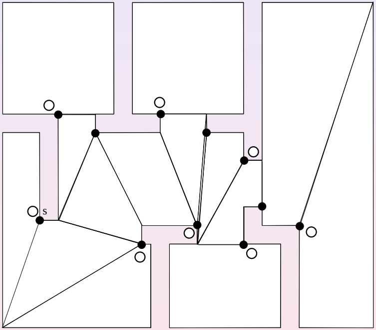
Sparse point set for deployment without connectivity



Sparse point set for deployment without connectivity



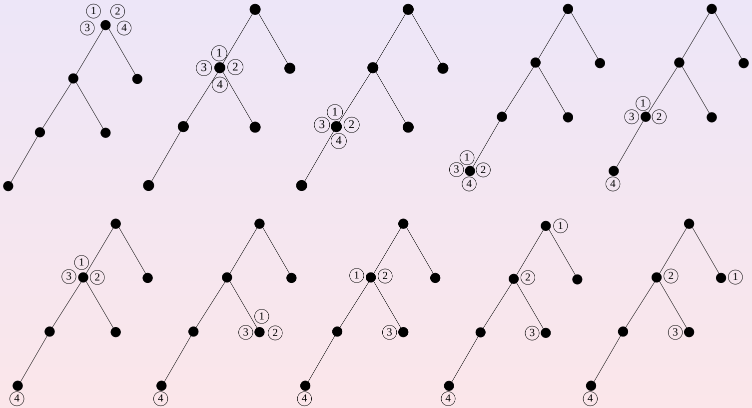
Sparse point set for deployment without connectivity



Every point in the kernel "owns" at least two quadrilaterals or four triangles
Total number of triangles is $n-2$

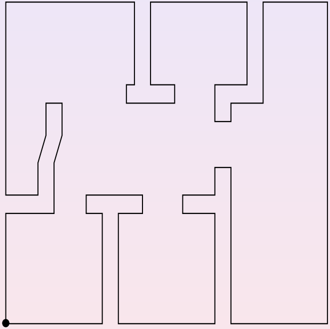
Therefore, number of points in the kernel is $n/4$.

Depth-first deployment



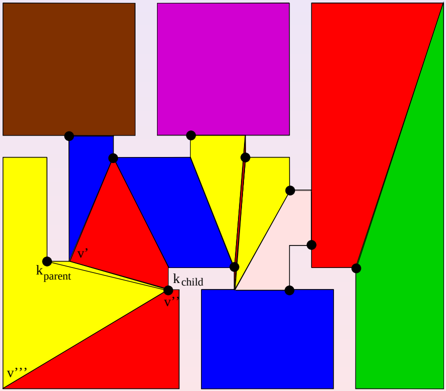
Assume: (i) Each node is a star-shaped set; (ii) Sets corresponding to non-leaf nodes are composed of a union of quadrilaterals equal in number to the number of children

Depth-first deployment



Depth-first deployment in general simply connected environments

Local navigation and distributed information processing



- Straight line paths between adjacent nodes
- Required memory: $\mathcal{M}_i : \{p_{\text{parent}}, p_{\text{last}}, g_1, g_2\}$
- After moving from k_{parent} to k_{child}, k_{parent} is added to the beginning of list p_{parent}, (v', v'') is added to list g_1, (v'', v'') is added to list g_2 and $p_{\text{last}} := k_{\text{parent}}$
- After moving from k_{child} to k_{parent}, the first elements of p_{parent}, g_1 and g_2 are deleted and $p_{\text{last}} := k_{\text{child}}$

Main results

Connected deployment

1. If # agents < cardinality of the sparse kernel point set,
   then in finite time each agent comes to rest at a unique kernel point
   else in finite time every kernel point contains an agent at rest
2. $\lfloor \frac{n}{4} \rfloor$ agents are always sufficient and occasionally necessary for the task

Deployment without connectivity

1. If # agents < cardinality vertex-induced tree,
   then in finite time each agent comes to rest at a unique node
   else in finite time every node contains an agent at rest
2. $\frac{n-2}{2}$ agents are always sufficient and occasionally necessary for the task

Outline

Conclusions

Summary

- distributed algorithms to achieve coverage in nonconvex orthogonal environments
- number of agents required is optimal in the worst case
- robustness to agent failures and changing environments

Future directions

- environments with holes
- 3D scenarios
- other notions of optimality: time taken, other complexity measures other than the number of vertices