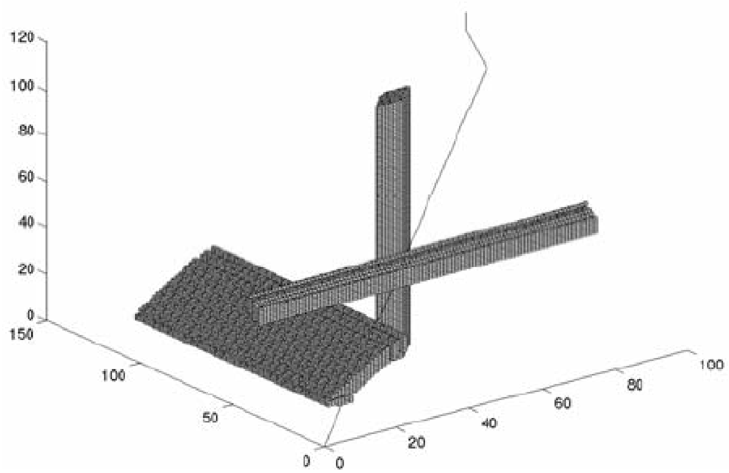
Multi-Robot Path Planning



Objectives

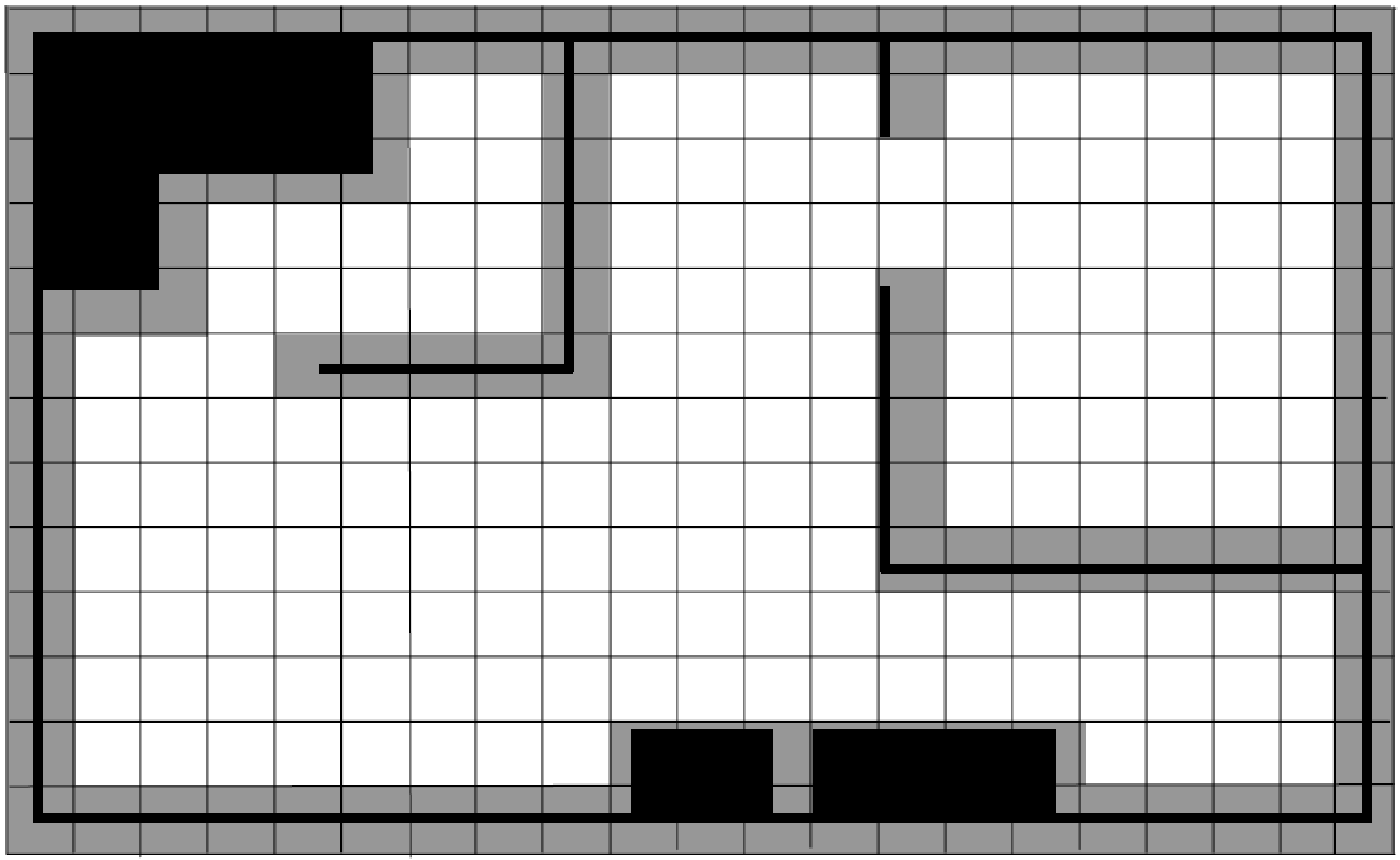
- Understand basics of path planning
- Multi-robot path planning
- Tight coordination in multi-robot teams

Navigation and Path Planning

- Navigation is a fundamental capability of autonomous mobile robotics

- Primary functions of navigation:
  - Where am I going?
    - Usually determined by a human operator or a mission planner
  - What’s the best way to get there?
    - The Problem of path planning: qualitative and quantitative
  - Where have I been?
    - Map making
  - Where am I?
    - Localization: relative or absolute

Metric or Layout: Grid Representation



Metric Path Planning

- Graph-based planner
  - Classical approach: A*
    - Plan once, and then reactively execute
  - Problems: lack of opportunistic replanning
    - Solution: Extension from A* to D*

Brief Introduction to A*

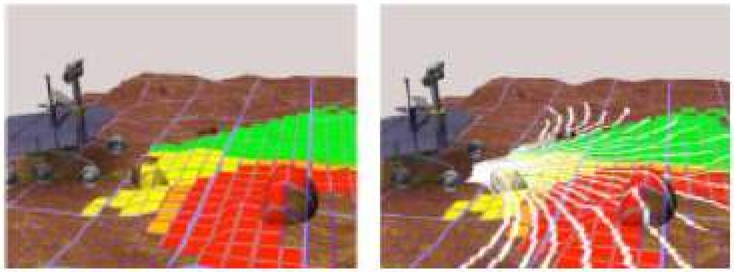
- A* is a classical graph based planner for finding path between initial node and goal node

- A* is able to prune off paths that aren't optimal and thus reduces the search tree size

- Basic idea: $f(n) = g(n) + h^*(n)$
  - g(n) is the actual path cost from initial node to node n
  - h*(n) is the estimated path cost from node n to the goal node
  - when h*(n) is ≤ true path cost from node n to goal node, h*(n) is called admissible, and A* is optimal

Extension: from A* to D*

- D*: initially plans path to goal just like A*, but plans a path from every position to the goal
  - Solve "all pairs of shortest path"
  - Dynamic A*
  - Planning with free space assumption
- In D*, the estimated distance, h*(n), is based on traversability



Calculate traversability using stereo cameras; can also manually mark maps

- Then, D* continues replanning, by updating map with newly sensed information

Multi-Robot Motion Planning: Background

- Objective: enable robots to navigate collaboratively to achieve spatial positioning goals
- Motion planning in dynamic environment with moving obstacles is NP-hard
- Simple reactive motion planning strategies cannot guaranteed to be deadlock free and to converge
- Previous results either obtain optimal solutions through centralized and exhaustive computing, or achieve distributed implementations without considering optimization issues
- Outdoor environment is more challenging with terrain features and requires online re-planning

“A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots”

by Guo and Parker, *Proceedings of IEEE International Conference on Robotics and Automation*, 2002

Decentralized Motion Planning

- Each robot plans its own path independently
- A coordination diagram is constructed based on collision checks among all robot paths
- A search algorithm is executed on the coordination diagram to generate a velocity profile which minimizes a global performance function

Multi-Robot Motion Planning: Contributions

- Approach to 3D multi-robot motion planning:
  - Distributed
  - Capable of outdoor environment and real-time re-planning
  - Uses global performance measurement for minimization
  - Uses D* search (Stentz, ICRA'94) to facilitate local search

Assumptions for Distributed and Optimal Motion Planning

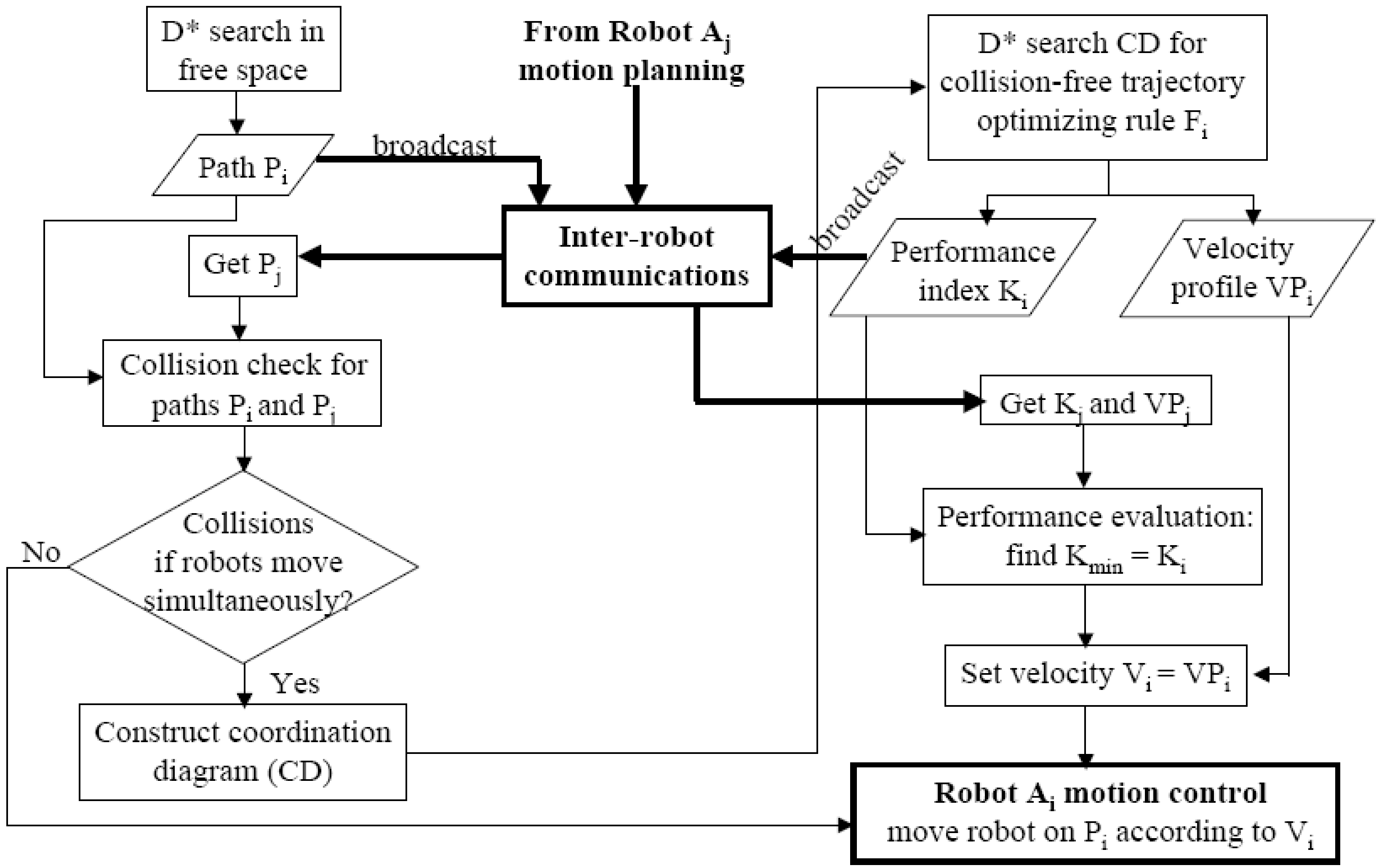
- Premises:
  - Each robot has an assigned goal, and knows its start and goal
  - Pre-defined map available that defines terrain elevation and traversability on a grid representation of the terrain
  - Onboard sensors detect discrepancy, and revise map online
  - Communication devices broadcast messages
  - Robots move at constant fixed speeds
  - Robots switch instantaneously between fixed speed and halting

MR Motion Planning Problem

- Based on assumptions 1-7, find a sequence of traverse states for each robot A_i moving from its start position S_i to its goal position G_i, without collisions with static obstacles and each other, while minimizing the following global performance index:

$$\Gamma = \gamma_1 \max(T_1, T_2, \ldots, T_N) + \gamma_2 \sum_{i=1}^{N} I_i \qquad (1)$$

Multi-Robot Motion Planning Algorithm for Robot A_i



Step 1: Path Planning

- D* search in free space to produce optimal path P_i for each robot from the start to the goal minimizing a cost function based on obstacles, distance, slope and turning

$$f_{pp} = \rho + \alpha_1 d + \alpha_2 s + \alpha_3 t$$

- D* returns an optimal path that avoids static obstacles, and is the shortest, flattest, smoothest possible path if one exists

- Outputs: a sequence of waypoints based on the map resolutioin

Step 2: Collision Check

- Generated paths communicated to all robots
- Collision check produces a set of collision regions
  - Represented by sets of (x, y) pairs at which path intersections occur

- Coordination diagram (CD) constructed by:
  - Mapping each path to a 1-D trajectory based on path length
  - N-Dimensional coordination space is defined as:
    - $S = S_1 \text{ x } S_2 \text{ x } \ldots \text{ x } S_N$, where $S_i = [0, l_i]$
  - S_i denotes the set of points that place the robot along the path, assuming robot moves at a constant speed
  - Collision regions marked as obstacles in coordination diagram

Steps 3 and 4

- Step 3: Velocity planning
  - Search in CD will find a non-decreasing curve that connects (0, 0, …, 0) to $(l_1, l_2, \ldots, l_N)$
  - D* search in CD, minimizing cost function of collision region, distance, idle time and penalty for giving way

$$f_{vp} = \varrho + \beta_1 d + \beta_2 t_{idle} + \beta_3 p$$

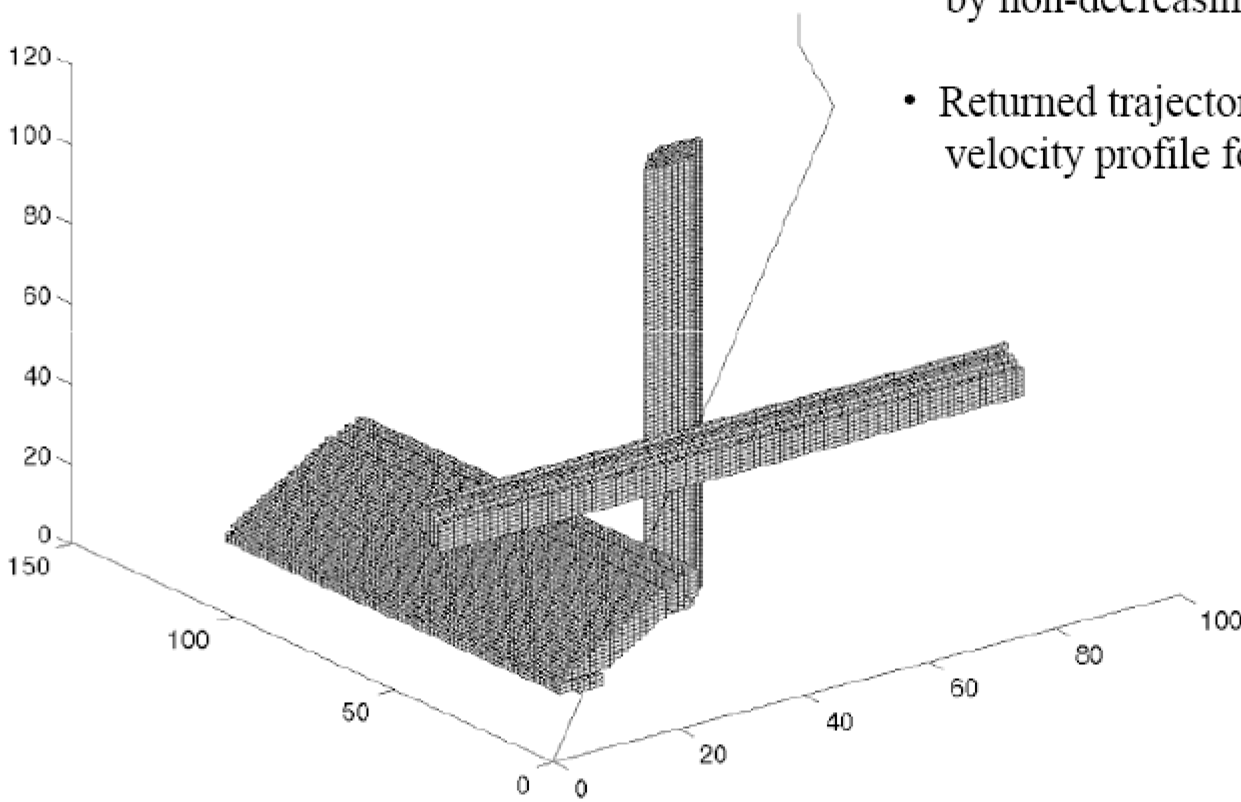
  - Produce velocity profile VP_i and performance index K_i
  - Communicate VP_i and K_i across robots

- Step 4: Global performance evaluation
  - Find the minimal K_i, select corresponding VP_i as the optimal solution for velocity

Search Result in Coordination Diagram

For 3 robot path coordination:

- Coordination diagram parameterized by non-decreasing path length
- Returned trajectory: interpreted as a velocity profile for each robot



Shaded areas denote intersection regions

Results of Multi-Robot Path Planning

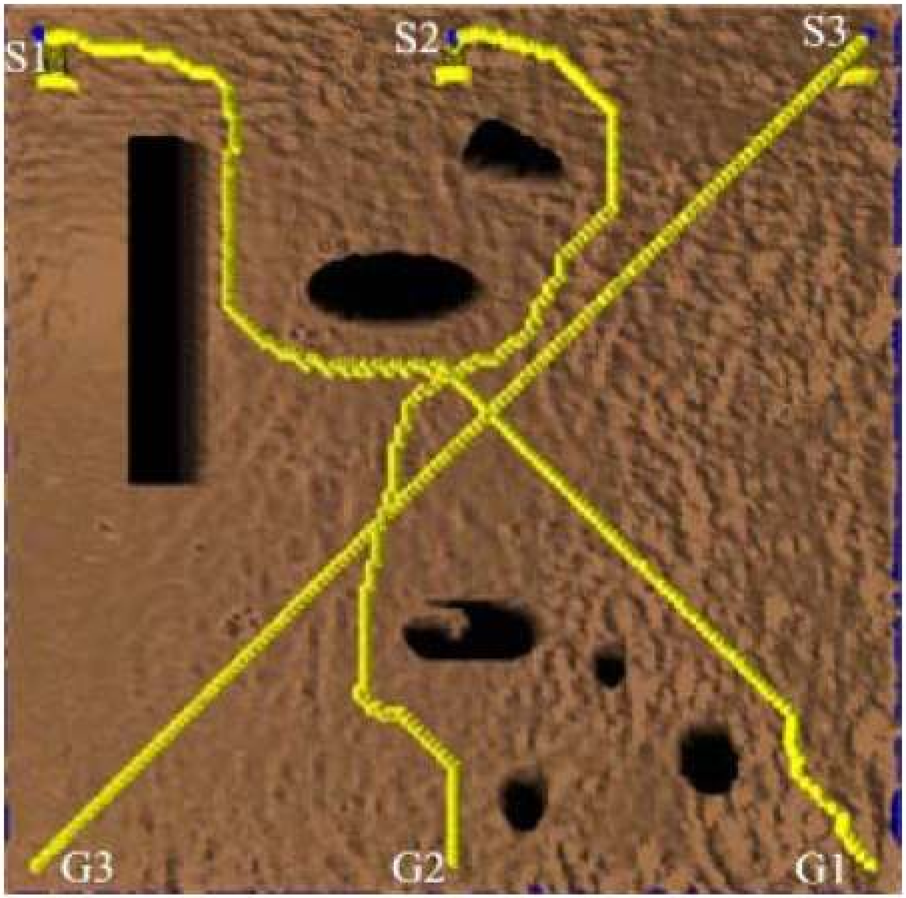


Fig. 3. 3D simulation results in an outdoor environment with a Mars-like terrain. Black areas are untraversable terrain. S1,S2,S3 denote the start positions of each robot respectively, and G1,G2,G3 denote the goal positions.

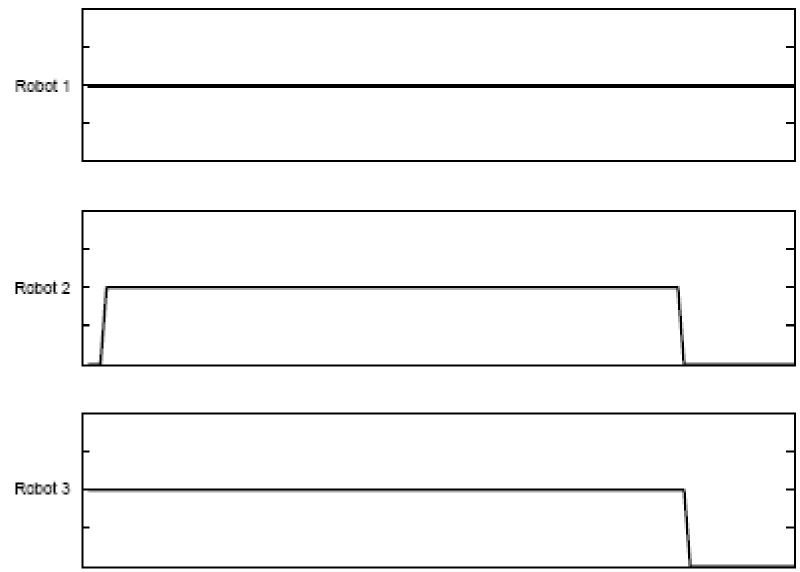
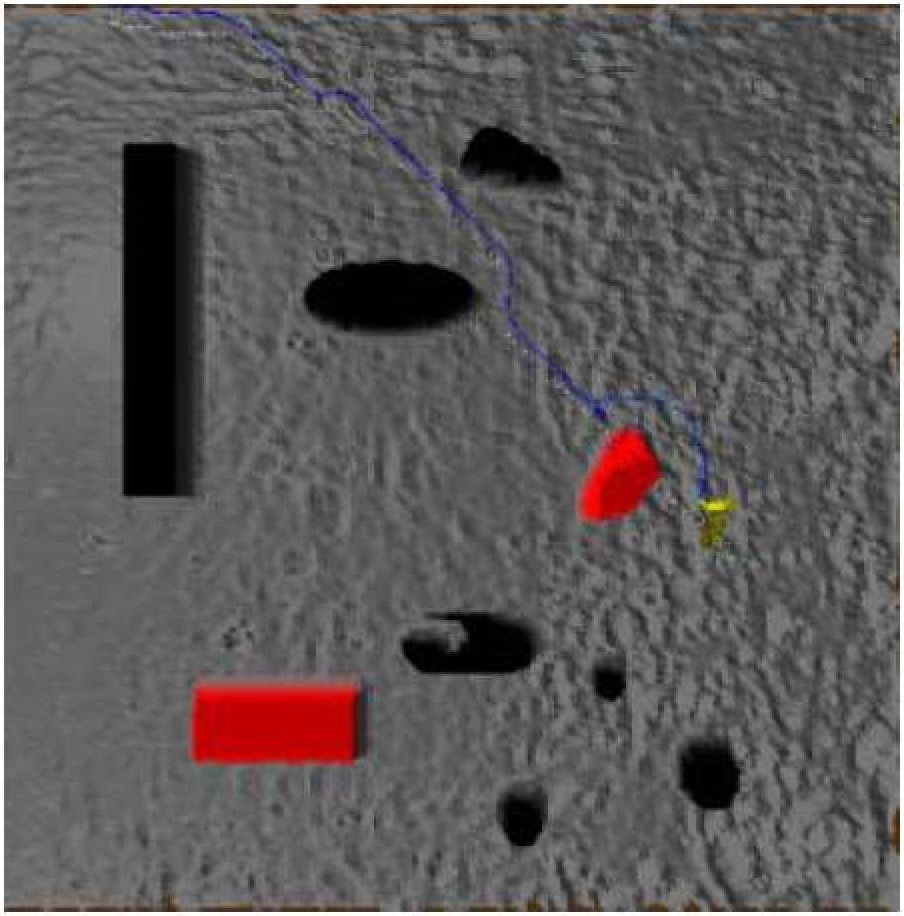


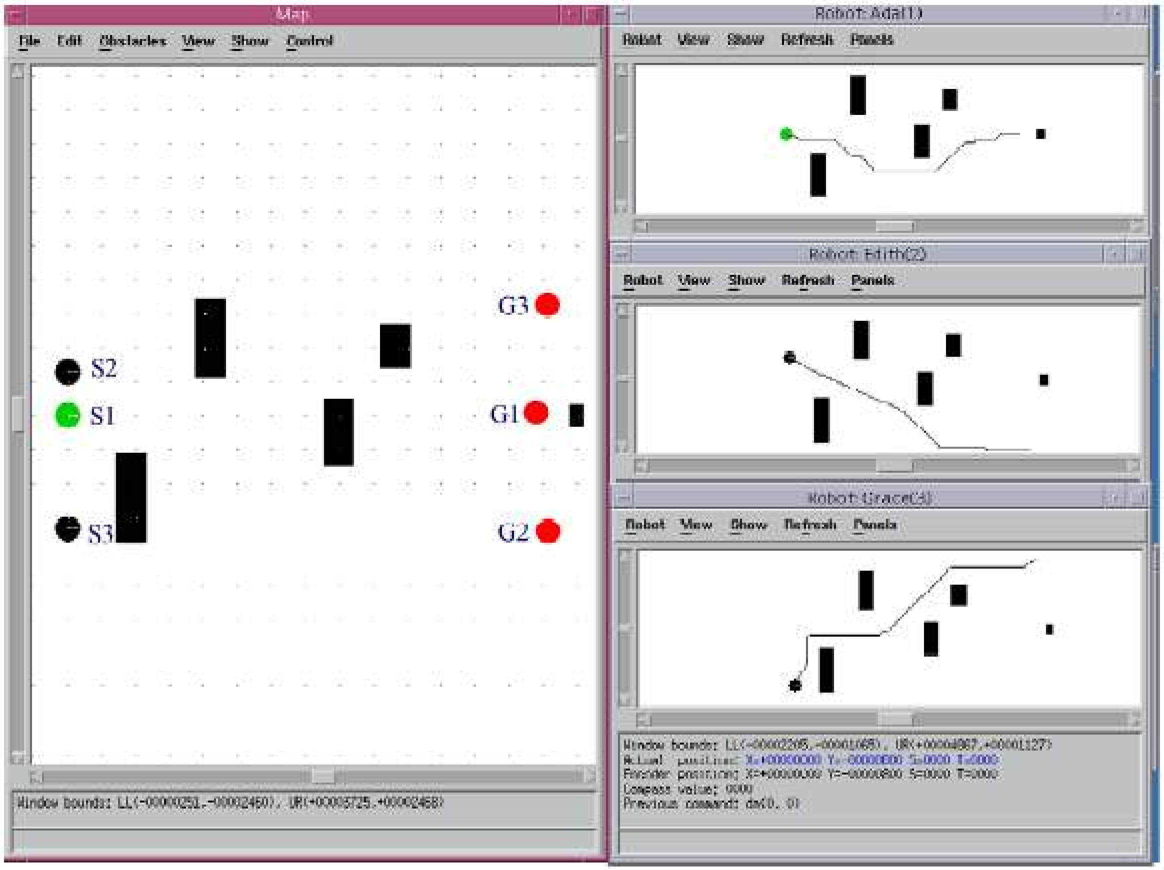
Fig. 4. Velocity profile of 3D simulations.

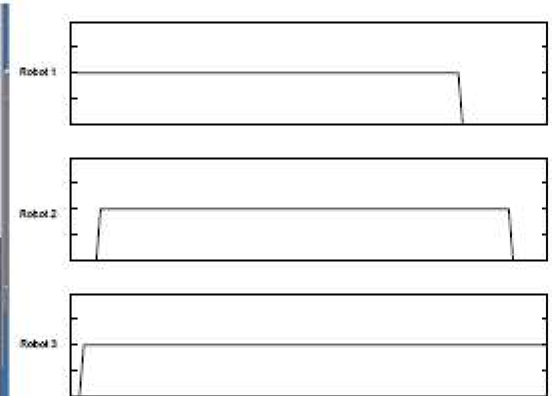
3D Simulation in Mars-Like Terrain

- Online re-planning function (red obstacles not in prior map)



Nomad 200 Indoor Robot Simulation





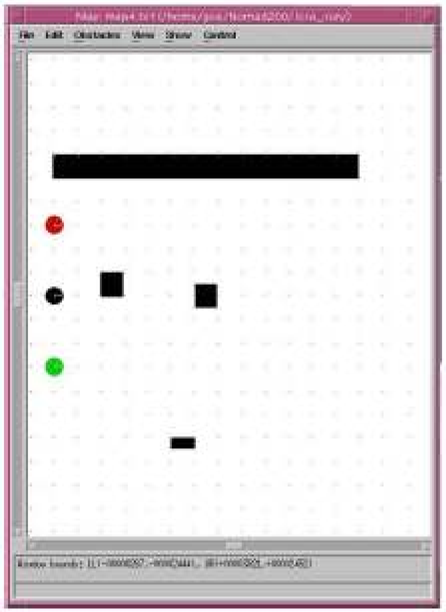
Indoor paths and velocity profile

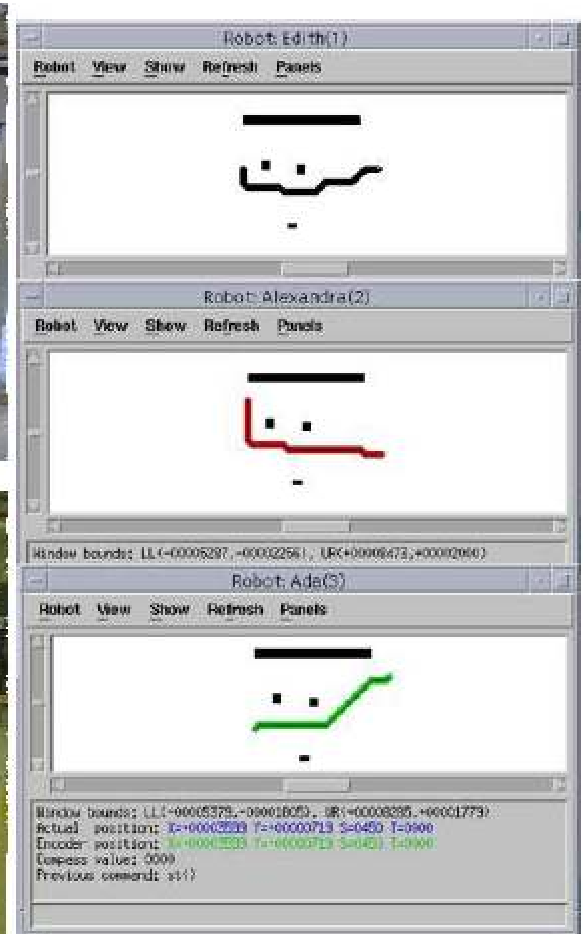
Nomad 200 Indoor Robots Experiments

- Problems met in real run:
  - Localization errors
  - Motion uncertainties
    - Robot does not take equal unit time to track a unit distance
    - Robot does not switch instantaneously between moving and stopping

- Robustness design
  - Safety margin defined in path searching for localization errors
  - Safety margin defined in velocity planning for motion uncertainties

Nomad 200 Indoor Robots Experiments





Pre-defined map,
Robots at run,
Encoder trajectories.

Summary of Motion Coordination Research

- Many issues studied by the field:
  - Multi-robot path planning
  - Traffic control
  - Formation generation
  - Formation keeping
  - Target tracking
  - Target search
  - Multi-robot docking

- Approaches are usually specific to given application

Hopelites: A Market-Based Framework for Planned Tight Coordination in Multirobot Teams

By Kalra, Ferguson and Stentz, ICRA 2005.

Presented by Kushal Patel

Tasks Requiring Coordination

- Address tasks that have constraints between robots, requiring complex coordination between team members
  - Gallery monitoring
  - Security sweep
- Mechanisms:
  - Tight coordination: robot A considers state of robot B at a high frequency when selecting its actions
  - Planned coordination: at some time t, robot A anticipates the interaction with robot B at a later time t’

Hoplites

- A market framework that consists of “passive” and “active” coordination mechanisms
  - Passive coordination:
    - Robots quickly react to each other’s actions and influence each other implicitly
  - Active coordination:
    - Robots influence each other explicitly by buying and selling complex team plans over the market

Market Framework

- Distributed Market economy
  - Robots gain revenue for completing tasks
  - Robots also consume resources
  - Robots trade tasks through auctions and negotiations to win tasks that generate the greatest profit (revenue – cost)

  - A robot can plan a more cost-effective task distribution for part of the team, bid on the tasks, and use the cost savings between the two distributions to purchase team members' participation → a better team solution

Passive Coordination

- Robot generates a set of plans and estimates the profitability of each plan
- It then broadcasts its most profitable plan to its teammates
- Teammates reevaluate their current plans

- Effective in environments where the correct actions are obvious to all robots

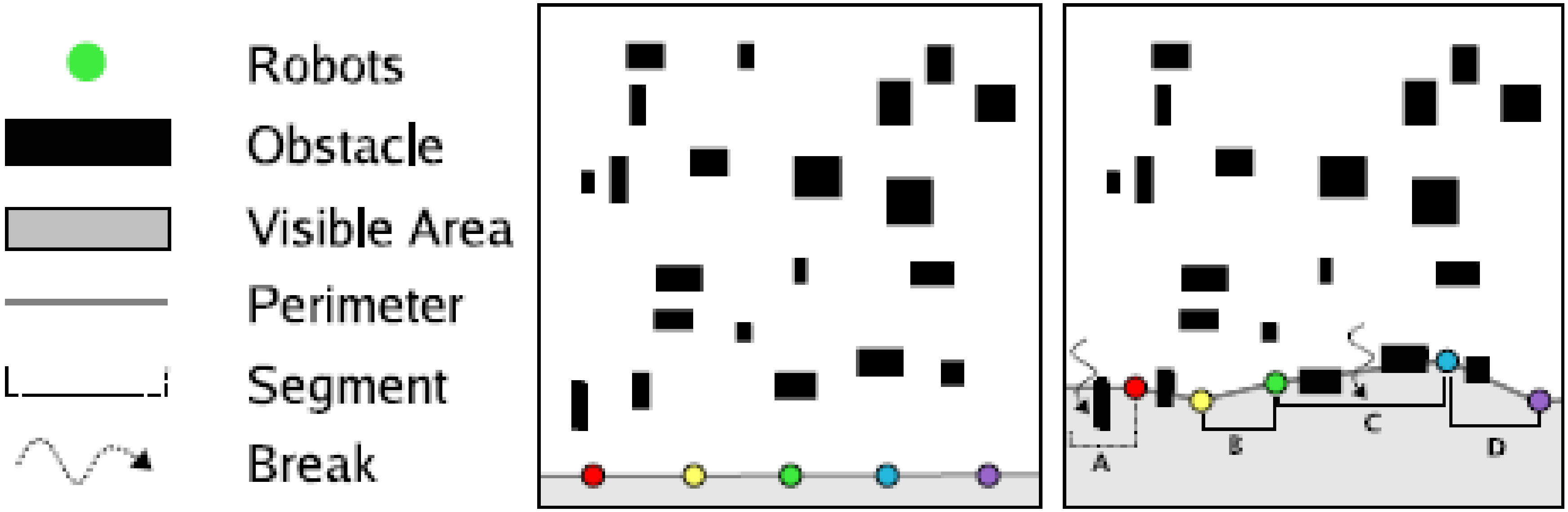
Active Coordination

- Difficult situation may require teammates' complex coordination (high-risk)
- Passive coordination does not guarantee its teammates' actions
  - Robots favor low-risk ones that are less profitable
- Robot develops a team plan that consists of actions of teammates
  - Makes sure that (Profit(P_team) – Profit(p_max)) is greater than cost of team plan (q1+…+qn)
  - Costs: others will abandon their existing plans
- If teammates accept, they are bound by contract to complete their portions

More Framework Details

- Choosing coordination strategies:
  - A robot passively coordinates until it expects that its best plan will be less profitable due to constraint violations;
  - It then actively coordinates
- Planning
  - To develop a team plan, a planner must tractably search joint action spaces
- Commitments between teammates
  - Robots are bound to a team plan except when they can negotiate breach-of-contract terms

Security Sweep Domain

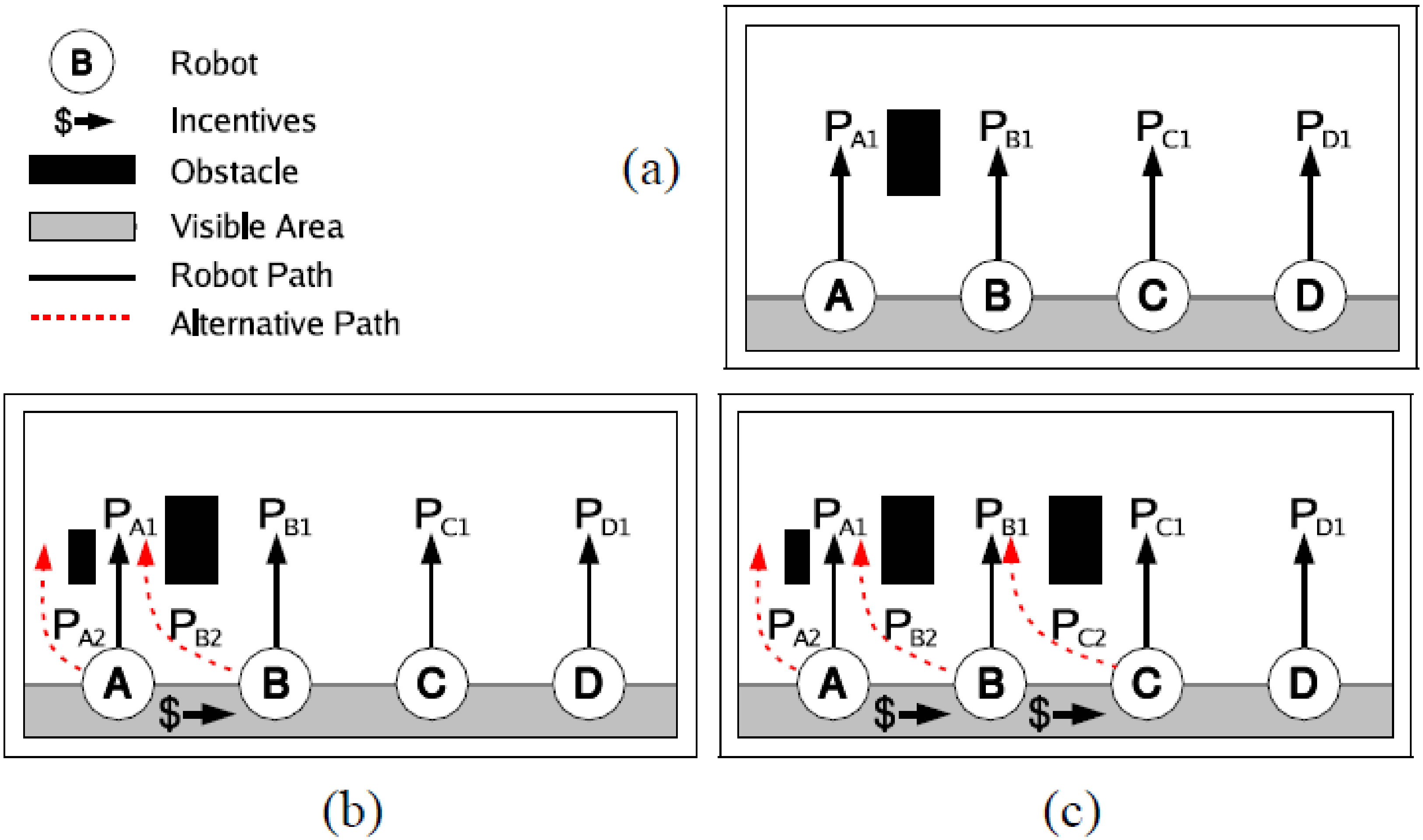


(a) (b)

Robot Revenues and Costs

- Each step in +y direction generates revenue proportional to X_max/n
- A robot incurs costs proportional to amount of time it takes and distance it travels
- A robot is penalized for breaks in the two segments adjacent to it
  - Penalty for each step that breaks a previously-secure segment or fails to repair one

Example



B
Robot
$→
Incentives
Obstacle
Visible Area
Robot Path
Alternative Path
(a)
P A1
P B1
P C1
P D1
A
B
C
D
P A2
P B2
P C2
(b)
(c)

Hoplites Enables Efficient Coordination

- Planning occurs through a series of linked plans (P_{AB} and P_{BC}), a single agent doesn’t have to plan for all
- Utility of a plan is evaluated locally, thus more accurate
- Robots can concisely transmit costs and benefits of a collection plans by combining a series of price quotes

Other Frameworks

- MVERT:
  - A robot first estimates the next action of every other team member
  - It then chooses the most valuable action given the expected team contributions
- P-MVERT:
  - Allow extended planning
  - A robot generates a set of candidate plans for itself and its teammates
  - It chooses a plan with highest expected profit given a uniform distribution over the set of teammates’ options

Other Frameworks

- PC-MVERT
  - Robots communicate their plans
  - Robots use their teammates’ intended actions to evaluate and select their own plans
  - Identical to passive coordination in Hoplites

Simulation Details

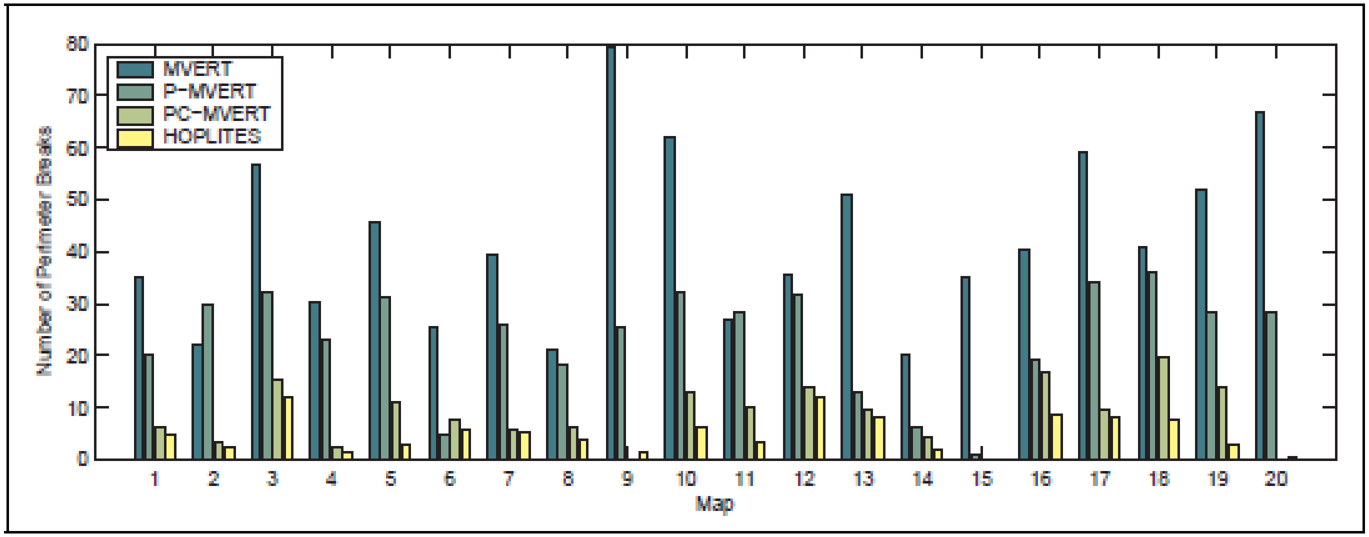
- 5 simulated robots in 200x200 units of environment
- Twenty randomly placed obstacles, ranging from 5x5 to 15x15 units
- Revenues and costs:
  - $500 penalty for breaking perimeter
  - Charge $5 per unit of time required to traverse environment and $5 per unit traveled
  - Receive $40 for each unit it moved towards the far end of environment

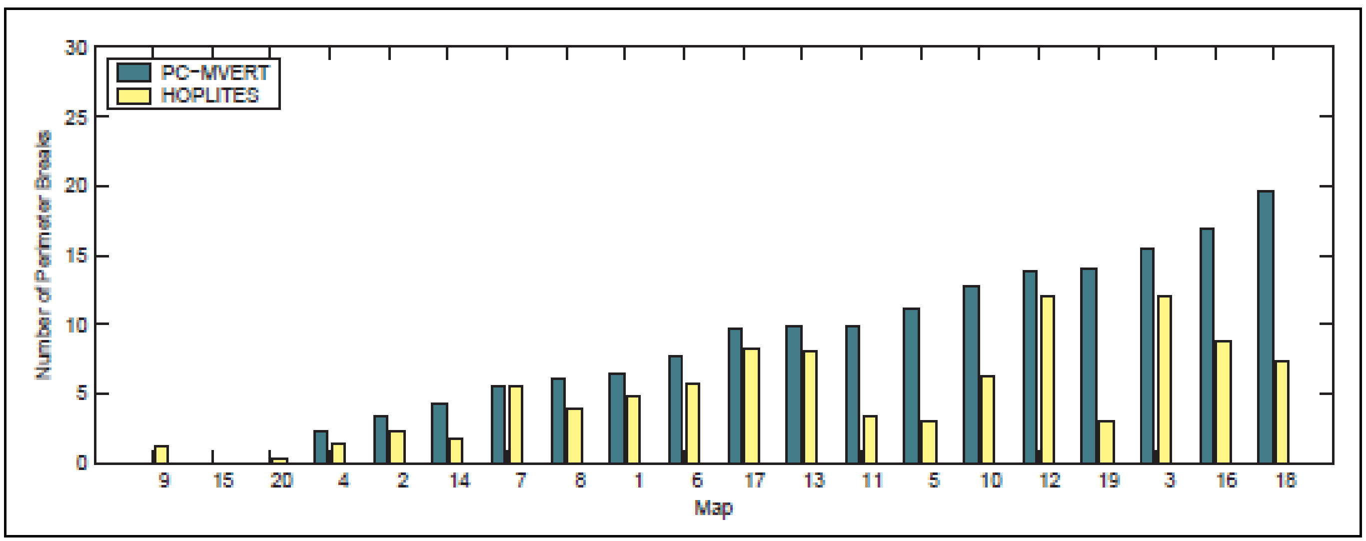
$$
\begin{array}{ll}
p = 40(y_n - y_1) & \text{Revenue} \\
\quad -500 \sum_{i=0}^{n-1} Broken(x_i, y_i, x_{i+1}, y_{i+1}) & \text{Penalty} \\
\quad -5 \sum_{i=0}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} & \text{Distance} \\
\quad -5(t' - t) & \text{Time}
\end{array}
$$

Results

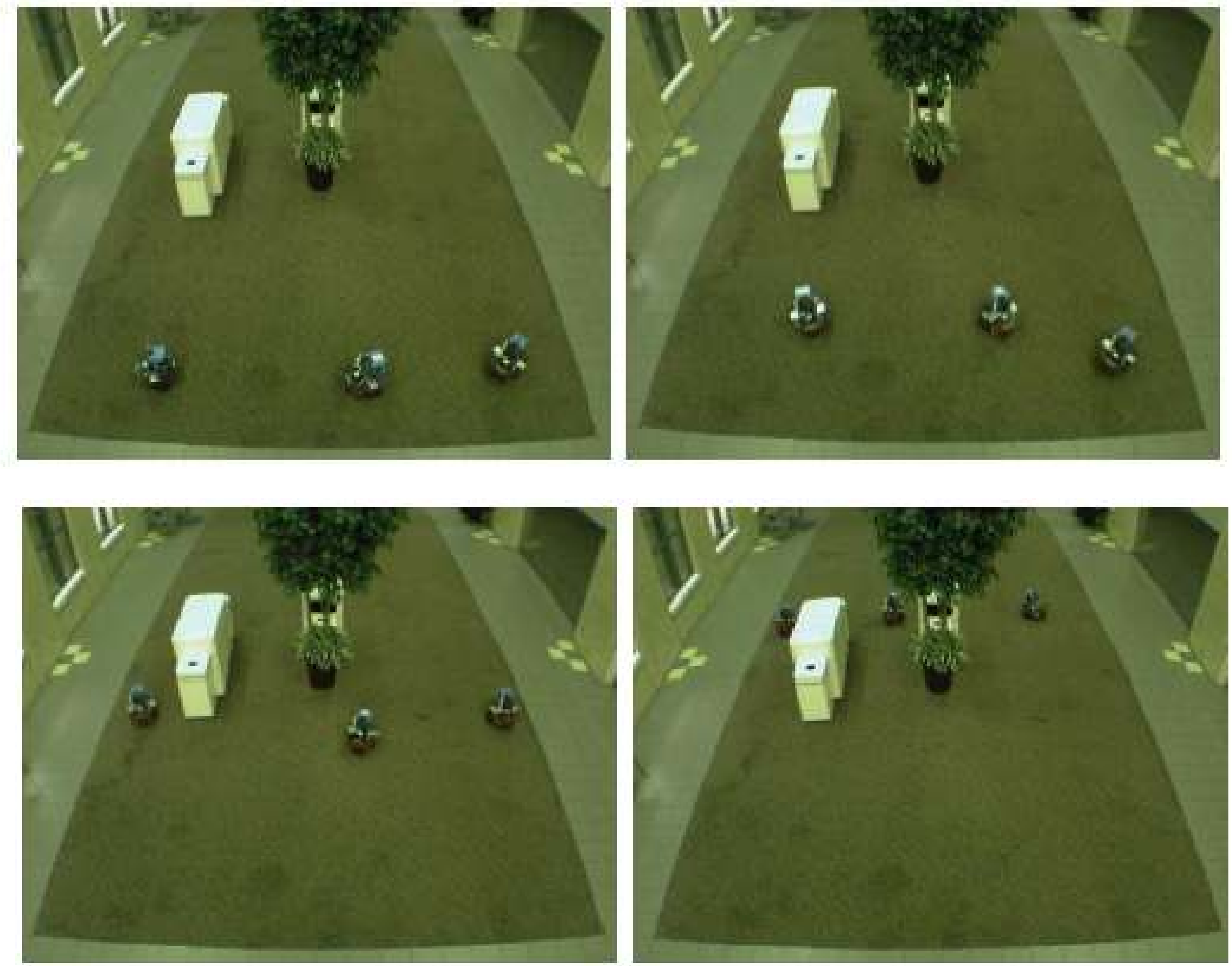
Algorithm	Runs	$\overline{B}$	$\mathrm{SEM}_{\overline{B}}$	$\mathrm{Rel}_{\overline{B}}$	$\overline{T}$	$\mathrm{SEM}_{\overline{T}}$	$\mathrm{Rel}_{\overline{T}}$	$\overline{D}$	$\mathrm{SEM}_{\overline{D}}$	$\mathrm{Rel}_{\overline{D}}$
MVERT	963	41.9	0.43	–	12.1	0.29	–	195.8	0.5	–
P-MVERT	973	22.4	0.51	0.54	29.2	1.72	2.41	200.4	0.53	1.02
PC-MVERT	957	8.6	0.20	0.39	20.1	0.36	0.69	198.8	0.54	0.99
Hoplites	966	5.5	0.15	0.63	28.9	0.90	1.44	208.6	1.44	1.05

More Results





Physical Experiments



Conclusions

- Hoplites outperforms competitors
- Hoplites outperforms the most in complex environment
- Security sweep domain requires planning
- Both planning and communicate improve performance of MVERT
- Passive coordination alone may trap robots in local minima
- Increase in computation time
- Increase in distance traveled