Title
Mobile Robots and Sensor Network: Working Together

Authors
Batalin, Maxim
Sukhatme, Gaurav S

Publication Date
2003-10-10

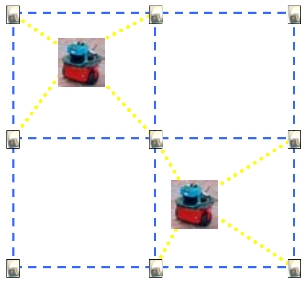
Mobile Robots and Sensor Network: Working Together

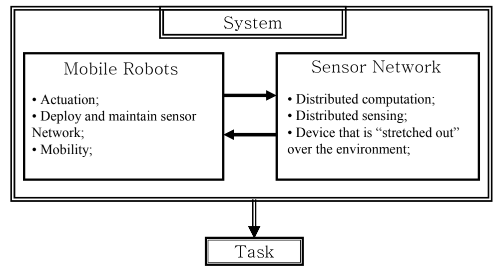
Maxim A. Batalin and Gaurav S. Sukhatme
Robotic Embedded Systems Lab - http://robotics.usc.edu/~maxim

Problem Definition

- Study the benefits of using a sensor network and mobile robots collaboratively
- Address following problems
  - Coverage
  - Navigation
  - Exploration
  - Network deployment
  - Network maintenance.

Sensor network provides distributed computation, communication and sensing for mobile robots. Robots, deploy and maintain the network.





Efficient Coverage/Exploration, Network Deployment and Maintenance

Problem Description: coverage/exploration of an unknown environment

Dynamic coverage: The environment is large enough (and possibly dynamic) that constant motion by the robot is needed to cover the environment

The coverage problem is the maximization of the total area covered by the robot's sensors

Applications: Tracking unfriendly targets (e.g military operations), demining or monitoring (e.g. security), and urban search and rescue (USAR) in the aftermath of a natural or man-made disaster e.g. building rubble due to an earthquake or other causes).

Assumptions:

- Neither a map, nor localization is used in a shared frame of reference;
- Our algorithm is based on deployment of static, communication-enabled markers into the environment by the robot;
- We assume the number of markers available for drop-off is unlimited;
- We assume that each marker being dropped off is capable of simple computation and communication;
- We do not assume that markers need to be retrieved;

Graph Analysis

We modeled our approach as a Graph Coverage algorithm. We compared the performance with Random Walk ($O(n^2)$)and Depth-First Search($O(n)$). The following conjectures were induced as part of the analysis:

Conjecture 1: The asymptotic cover time of our algorithm is less than $O(n \log n)$.

Conjecture 2: Our algorithm produces a map of the environment in asymptotic time faster than O(n logn).

Approach

Our algorithm uses two entities: the markers and the mobile robot. The task of each marker is to recommend a locally preferred direction of exploration for the robot within its communication range. Each marker has a state associated with the four cardinal directions (South, East, North, West). For each direction, the marker maintains a state (OPEN or EXPLORED) and a counter. The robot combines this suggestion with its local sensing and makes a decision as to which direction to pursue next. The robot deploys a new marker into the environment only if it does not receive a direction suggestion from any of the deployed markers. Approach also maintains the network - repairs it if some nodes were damaged, expands the network if hidden areas are discovered.

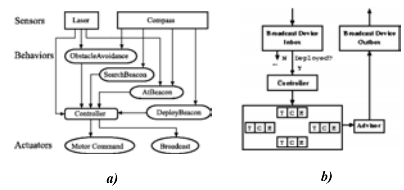


Figure 1: a) System Architecture showing Robot Behaviors;b) Beacon Architecture;

Simulation Experiment

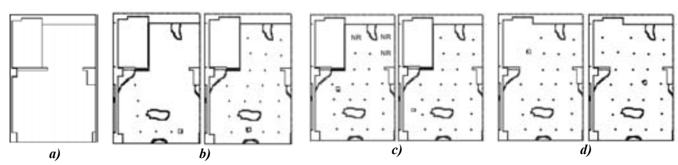


Figure 2: a) Environment before perturbations; b) Phase 1 of the experiment - Environment was perturbed. Robot explores the environment, while deploying sensor network and covering the environment; c) Phase 2 of the experiment - Nodes marked as NR are damaged and need to be replace; d) Phase 3 of the experiment - Environment was perturbed and a new 'hidden' room was unveiled.

Probabilistic Navigation

Problem Description: One of the fundamental problems in robotics is navigation. Navigation is the problem of getting from point A to point B while minimizing a cost function (e.g. travel time).

Approach: While sensor network is deployed by an algorithm described above transition probabilities *P(s'| s, a)* for every deployed node and every direction can be recorded. Where *P(s'|s, a)* is the transition probability of arriving to node *s'* if an action *a* was taken at node *s*. Given transition probabilities a Navigation Field can be computed for any *'goal node'*.

Distributed Value Iteration:

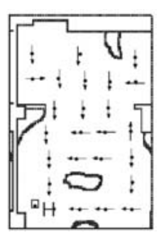
In order to compute a suggested direction of maximum utility, every node in the network updates its utility locally, queering neighbors for their corresponding utilities. The update equations are:

$$V_{t+1}(s) = C(s,a) + \max_{a \in A(s)} \sum_{s' \in S-s} P(s'|s,a) * V_t(s')$$

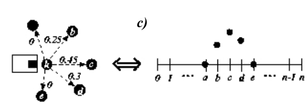
$$\pi(s) = \arg\max_{a \in A(s)} \sum_{s' \in S-s} P(s'|s,a) * V(s')$$

Where - *V* is the utility value, *C(s, a)* is the cost associated with an action, *P(s'|s, a)* is the transition probability of arriving to node *s'* if an action *a* was taken at node *s*, *π(s)* is the the policy, or direction that the node *s* is going to suggest for thee robot in the vicinity.

Figures an Graphs



a)



c)

d)

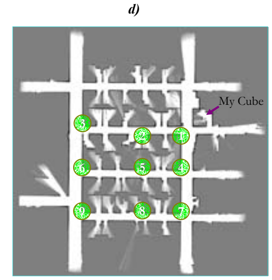
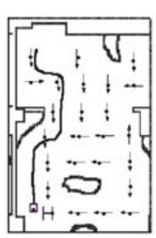
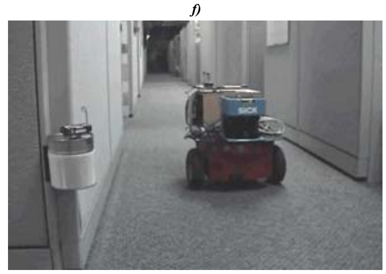


Figure 3: a) Navigation Field produced; b) Example of the path taken; c) An example of a discrete probability distribution of node k for direction (action) "East"(i.e. righ); d) Experimental environment with 9 nodes; e) Path taken by the robot from node 1 to node 9; f) Mobile Robot and a Sensor Node during a real world experiment in the environment in d);



b)

f)



e)