Distributed algorithm for robotic network self-deployment using wireless signal strength measurements

The University of Tokyo ○ Miyagusuku Renato, Yamashita Atsushi, Asama Hajime

Mobile robots with wireless capabilities can enable network connectivity over large areas by relaying wireless signals from a ground station. Our goal is, for such robotic networks, to enhance current teleoperated robots' ability to perform reconnaissance or assist human first responders on victims' search and rescue operations. On these missions, uninterrupted communications between teleoperated robots and their human operators are essential. In order to maximize the teleoperated robots' working area, the robotic network has to deploy itself, in an unknown and potentially hazardous environment, spreading as much as possible without losing network connectivity. Therefore becoming an area coverage problem. In this paper, we present a distributed algorithm that enables simple mobile robots with no advanced self-localization capabilities to create a robotic network and self deploy. The proposed method employs a behavior-based distributed algorithm mainly based on wireless received signal strength measurements and relative locations between neighboring robots - which are obtained from particle filters. Real wireless signal strength and odometry sensors' data were collected and used to validate our approach.

1. Introduction

Robots can be used to access unsafe locations and support victim search or aid first responders on victim rescue operations; unfortunately, limited range/unreliable communications between deployed robots and base stations prevent rescue robotic system from potentially performing adequately on real scenarios[4]. Furthermore, is on these disaster-stricken scenarios that communications are notably challenging, as wired networks most likely malfunction and collapsed walls and debris hinder wireless signals. The work herein presented intends to solve this problematic situation by developing a multiagent system composed by several mobile robots with wireless capabilities, which can expand wireless coverage at harsh communication environments or large areas.

We employ a distributed algorithm that, opposed to centralized ones[5], efficiently scales with respect to the number of robots and requires no explicit communications (differing from work like Ludwig and Gini[2]). In essence, our method utilizes a behavior-based approach[3] that employs wireless strength signal measurements, odometry and bearings to closest neighbors - estimated by particle filters.

2. Deployment of network nodes

We assumed robots have 802.11-compliant wireless network interfaces, basic obstacle avoidance and odometry sensors; and under this assumptions developed a behavior-based system that can effectively spread this robots across a delimited area, without losing connectivity. Even though our approach considers that each robot executes the same program using only information sensed or inferred by itself, at the system-level the desired global behavior emerges. This software is ruled by two basic behaviors: Collision avoidance and Aggregation-dispersion.

Collision avoidance is straight forward: *If an obstacle is detected too close, get away from it*. It is in charge of safe navigation, therefore taking top priority. Aggregation-dispersion's basic idea is: *Spread away from neighboring robotic routers, but not too much that you can lose connectivity*. This is attained by generating virtual forces based on the robot's 4 closest neighbors. For each of these neighbors, if the received signal strength indicators (RSSI) is higher than a pre-established dispersion threshold (neighbor is too close) a virtual force opposite to the neighbor's direction is created; contrary, if the RSSI is lower than the aggregation threshold (neighbor is too far) the virtual force points towards the neighbor. At each time step all generated forces are added and the robot moves towards the resultant's direction. It's important to notice that in order to generate these forces bearing to neighbors are required; not possessing sophisticated localization sensors, bearing estimation can pose a challenge. To solve this, our method uses particle filters, which work by estimating the posteriori density of neighbors' positions using odometry and RSSI measurements.

The filter works by generating a fixed amount of samples, each with a random possible location of one neighbor. At each time step, after the robot has moved, odometry is calculated and the samples are virtually moved. At their new positions virtual RSSI values are calculated and compared with the actual RSSI values sensed. How closely related both measurements are, establishes the probability each sample has of being at the correct position. This probability is used to generate a new set of samples; and the process is repeated indefinitely. Although convergence cannot be guaranteed, simulations show that after certain amount of time convergence is achieved - bearing errors after 50 iterations were 0.0421 +/- 0.0484 rad.

For clarity's sake, Alg. 1 shows the deployment of network nodes algorithm's pseudo-code.

3. Experimental validation

For the experimental setup, a roomba 500 series and a laptop running Ubuntu OS, as shown at Fig. **1** were used as platform. We used Roomba's incremental encoders placed one at each wheel for odometry, and the RSSI obtained from the laptop's wireless card (in monitor mode and using the libpcap library) as sensors.

Algorithm 1 Deployment of network nodes algorithm

Calculate Odometry and acquire RSSI signals from neighbors
Update particle filters and calculate bearing estimates
If no obstacles near:
  For 4 closest neighbors calculate virtual forces
  Add virtual forces and move robot towards resulting force
Else: Avoid obstacle

Fig. 1: Experimental setup

For odometry error estimation, acquired data from encoders were compared with data from a calibrated motion capture system. The test consisted on making the roomba robot drive generating different polygons such as squares or octagons. After processing the data, a forward movement error of 4% and an angle rotation error of 0.8% were estimated. Figure **2** shows data obtained from one of these tests.

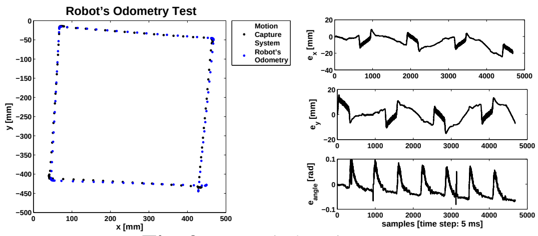


Fig. 2: Roomba's odometry

For Signal strength measurements error estimation data was taken along an standard hallway with plenty of interferences from walls, as well as several unrelated wireless networks operating at the same time. Readings were noisy even after filtering; therefore, a sensing error of 10 m was considered. Figure **3** shows data from one of the experiments.

In order to assess the effect of odometry and RSSI errors, particle filter algorithm simulations were performed. Setting odometry errors of 4% forward movement and 1% angle rotation, and RSSI errors equivalent to 10 m, the bearing error after 50 iterations, increased from 0.0421 +/- 0.0484 to 0.0984 +/- 0.0744 rad. Even though these errors drastically increased, as it is illustrated by Fig. **4**, for bearing errors as high as 0.75 rad (value significantly higher than 0.2474 rad - two standard deviation over the before mentioned expected bearing error) the system still has a high probability of retained an adequate connectivity.

4. Conclusions and future work

In this paper we have demonstrated the feasibility of using robotic routers to deployed and maintain a wireless ad-hoc networks. By using local behaviors, deployment of networks was achieved. The use of particle filters for bearing estimation using odometry and RSSI data was proposed, and its applicability to this specific problem was demonstrated through simulations. Finally, we collected real sensors' data using our platform and once errors were quantified, they were used to enhance the system's simulations, finding a recommended bearing error maximum value of 0.75 rad under which deployment without connectivity loss is quite successful; however, physical testing of the whole system remains as future work. It also remains as future work to develop a more explicit mechanism to guarantee network connectivity when mobile users go beyond the networks original coverage, when such event occurs, the network should re-arrange itself (position and topology) as to provide area coverage to section on the workspace that the mobile users intend to explore. Additionally, coverage under robotic routers' failure needs to be explored, for which ideas related to graph bi-connectivity [1] are being evaluated.

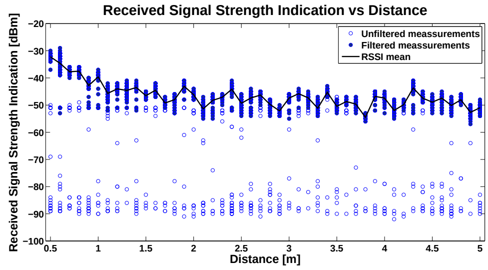
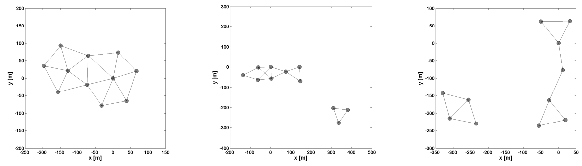
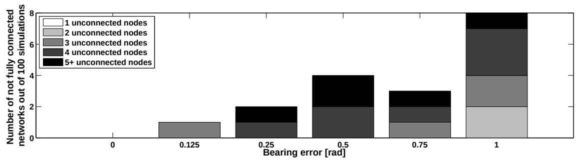


Fig. 3: Sampled RSSI data from an specific source, obtained by selecting only beacon sources with the desired source MAC address from all read packages.



(a) Full connection (b) 3unconnected node (c) 4unconnected node



(d) Not fully connected networks with respect to bearing angle errors. Total number of simulations 100.

Fig. 4: Network connectivity after deployment of 10 robotic routers with different bearing errors and 10 m distance error.

References

1) Mazda Ahmadi and Peter Stone. Keeping in touch: Maintaining biconnected structure by homogeneous robots. In *Proceedings of the National Conference on Artificial Inteligence*, volume 21, page 580. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
2) Luke Ludwig and Maria Gini. Robotic swarm dispersion using wireless intensity signals. In *Distributed Autonomous Robotic Systems 7*, pages 135–144. Springer, 2006.
3) Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.
4) Satoshi Tadokoro. *Rescue Robotics: DDT Project on Robots and Systems for Urban Search and Rescue*. Springer, 2009.
5) Onur Tekdas, Wei Yang, and Volkan Isler. Robotic routers: Algorithms and implementation. *The International Journal of Robotics Research*, 29(1):110–126, 2010.