CoCoA: Coordinated Cooperative Localization for Mobile Multi-Robot Ad Hoc Networks

Dimitrios Koutsonikolas Saumitra M. Das Y. Charlie Hu
Yung-Hsiang Lu C.S. George Lee
School of Electrical and Computer Engineering, Purdue University, USA
{dkoutson, smdas, ychu, yunglu, csglee}@purdue.edu

Abstract

Mobile robot teams are particularly suited to many application scenarios where infrastructure is unavailable or damaged. For example, mobile robot teams can be useful for exploration in remote regions or search and rescue after a disaster. Localization of individual robots in these teams is essential for enabling many applications or improving the robot's performance in particular tasks. However, in infrastructure-less application scenarios, conventional techniques for localization have many disadvantages such as cost, deployment time, inaccuracy and energy use. Thus, there is a need for a localization scheme that works in infrastructure-less scenarios and is low-cost, quickly deployable and energy-efficient while providing reasonable accuracy for the applications.

In this paper, we propose CoCoA, Co*ordinated* Co*operative* A*d-Hoc localization. In CoCoA only a subset of the robots in the mobile robot team are equipped with external localization devices (e.g. GPS, or laser rangers with SLAM). Subsequently, while robots perform their tasks, the subset of robots with localization devices help to localize other robots, avoiding the need for static landmarks to be deployed. This is achieved using a modified Bayesian inference-based localization algorithm previously proposed for localization in sensor networks. In addition, CoCoA coordinates this localization process using multicast to put wireless devices in sleep mode periodically which provides significant energy savings. Using detailed simulations and localization models calibrated from experimental data, we find that CoCoA is effective in reducing energy consumption while providing good localization accuracy.*

Index Terms - mobile robots, wireless communication, localization, cooperation, energy-efficiency.

1 Introduction

In many applications, mobile robot teams need to be deployed quickly in an ad-hoc manner and work without significant infrastructure support. Such scenarios arise when exploring remote terrains, in disaster relief or when the system targets low costs. In most such application scenarios, localization is a critical system service required for the mobile robot teams to operate effectively and efficiently. For example, in search and rescue operation, the location of a survivor needs to be indicated so that additional personnel can be dispatched to the area. In this paper, we explore how to provide a localization system for mobile robot teams that work in such infrastructure-less scenarios.

There are typically two extreme solutions to localization in such systems. On one extreme, all the robots in the mobile robot team could potentially be equipped with localization devices (e.g. GPS, laser rangers with SLAM) so that no cooperation is necessary. However, outfitting such devices on *all* robot nodes increases system costs and energy drain. On the other extreme, each robot in the mobile robot team could be provided with its initial position and deployed, after which it would rely on odometry to localize itself. However, this technique is known to suffer from accumulated localization errors due to the inherent difficulties in providing highly accurate odometry. The localization error is likely to be exacerbated by the uneven surfaces encountered in many application scenarios. Improving the performance of this technique would require more precise but costly odometry devices and there are limits to the accuracy possible even with costly devices.

In this paper, we propose the CoCoA system that takes a middle-ground between these two extremes. In CoCoA, only a subset of nodes in the mobile robot team is equipped with localization devices in order to reduce costs. Subsequently, CoCoA makes the robots cooperate in localizing each other using RF beacons through cheap commodity IEEE 802.11 wireless cards. Mobile robot teams in many cases already have such wireless devices for inter-robot or robot-to-operator communication and they can be reused for CoCoA as well. It is well known that wireless transmissions consume energy heavily. Thus, CoCoA also proposes to coordinate this beaconing in order to reduce the energy usage of the mobile robot team spent on localization. The coordination is achieved using the Mobile Robot Mesh Multicast (MRMM) [1] protocol which itself operates in an infrastructure-less environment. In summary, CoCoA provides a low-cost, quick way to deploy and localize multi-robot teams for infrastructure-less application scenarios. We evaluate CoCoA using detailed simulations and localization models calibrated from experimental data.

The rest of the paper is organized as follows: Section 2 describes the design of the CoCoA system, and the two components that constitute it: the cooperative localization and the energy-efficient coordination. Section 3 describes the experimental methodology. Section 4 gives the evaluation of our system. Section 5 summarizes the related work. Finally, in Section 6 we conclude the paper and discuss future work.

2 CoCoA Design

In this section, we describe the design of CoCoA. We first state our assumptions about the mobile robots in the team as well as the RF communication mechanism. We then describe the two key components of CoCoA: (1) the cooperative localization algorithm, and (2) the energy-efficient coordination mechanism.

2.1 Assumptions

We focus on an application scenario where many robots are used to form a mobile robot network. Each robot has a simple sensory ability and limited computational power. This makes it practical to build a large number of such robots. The communication among the robots is based on cheap open-license wireless communication (IEEE 802.11b) at 2.4 Ghz. This reflects our mobile robot testbed in which mobile robots are equipped with laptops and PDAs and use Orinoco Wavelan 802.11b wireless cards for communication. In addition, our testbed has 4 laser rangers that can be used through the laptop/PDA for robot applications. Thus, we assume that a subset of nodes are equipped with laser ranger receivers that provide localization through a SLAM (Simultaneous Localization and Mapping) algorithm [2]. The size of this subset is a tradeoff between cost and energy versus localization accuracy and we study this in the evaluation.

2.2 Cooperative Localization

The first key component of CoCoA is a cooperative beacon-based localization scheme. In CoCoA, robots that have localization devices transmit beacons while performing their tasks and robots that do not have localization devices receive these beacons to localize themselves. Whenever a robot without a localization device receives an RF beacon from another robot, it executes a localization algorithm, in order to refine its position estimate. This algorithm was first proposed by Sichitiu et al. [3] for localization in *static* sensor networks. It uses the Received Signal Strength Indicator (RSSI) for ranging and Bayesian inference to estimate the positions of the unknown nodes. We apply this algorithm to mobile robot nodes.

Before running the algorithm, an offline calibration phase is necessary, which is described in the next section. This phase constructs the PDF Table, which is stored at each node and maps every RSSI value to a Probability Distribution Function (PDF) versus distance. According to the algorithm, the robots with localization devices periodically broadcast beacon packets as they move in the deployment area. These packets contain the coordinates of the robot (x_B, y_B). When a robot without a localization device receives a beacon packet, it performs a lookup at the PDF Table and obtains the probability distribution function of the distance corresponding to the RSSI of the beacon packet. Using this function, the robot imposes the following constraint on its position estimation:

$$\begin{aligned} Constraint(x,y) &= PDF_{RSSI}(d((x,y),(x_B,y_B))) \\ \forall (x,y) &\in [(x_{min},x_{max}) \times (y_{min},y_{max})] \end{aligned} \tag{1}$$

where PDF_{RSSI} is the probability distribution function, $d((x,y),(x_B,y_B))$ is the Euclidean distance between the points with coordinates (x,y) and (x_B,y_B), and x_{min}, x_{max}, y_{min}, y_{max} are the bounding coordinates of the deployment area.

Bayesian inference is then applied and the new position estimate $NewPosEst$ is computed, based on the old position estimate $OldPosEst$ and the new constraint $Constraint$:

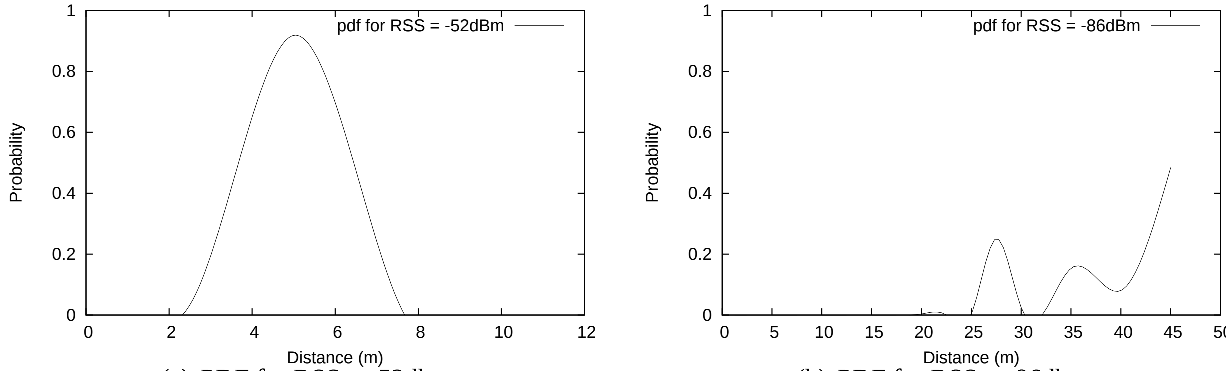
$$\begin{aligned} & NewPosEst(x,y) = \\ & \quad \frac{OldPosEst(x,y) \times Constraint(x,y)}{\int_{x_{min}}^{x_{max}} \int_{y_{min}}^{y_{max}} OldPosEst(x,y) \times Constraint(x,y)} \\ & \quad \forall (x,y) \in [(x_{min},x_{max}) \times (y_{min},y_{max})] \end{aligned} \tag{2}$$

The initial position estimate for each robot is initialized to a constant value, since in the beginning, a robot is equally likely to be in any position in the deployment area.

This process is repeated for each received beacon packet. Finally, if the robot has received at least three beacon packets, it uses the last position estimate PosEst to compute its best position coordinates $(\hat{x},\hat{y})$ as follows:

$$\begin{aligned} \hat{x} &= \int_{x_{min}}^{x_{max}} \int_{y_{min}}^{y_{max}} x \times PosEst(x,y)\,\mathrm{d}x\mathrm{d}y \\ \hat{y} &= \int_{x_{min}}^{x_{max}} \int_{y_{min}}^{y_{max}} y \times PosEst(x,y)\,\mathrm{d}x\mathrm{d}y \end{aligned} \tag{3}$$

Experimental Verification This algorithm assumes that, for each signal strength value, the probability distribution function of this value versus distance is Gaussian. In our outdoor experimental tests with our mobile robot equipment, we have found that this assumption correctly models the real world



(a) PDF for RSS = -52dbm (b) PDF for RSS = -86dbm

Figure 1. Probability distribution functions (PDFs) for two different Received Signal Strength values

for signal strength values up to -80dbm, which correspond to physical distances of up to 40 meters. One example of this function is shown in Figure 1(a) for RSSI = -52dbm. For distances larger than 40m, the noise in the signal strength measurements fluctuates due to multipath and fading, and the probability distribution function of the signal strength versus distance can no longer be approximated by a Gaussian, as shown in Figure 1(b) for RSSI = -86dbm. Note that this reflects our particular hardware and may not be general for all configurations. However, for most 802.11b cards which typically have a transmission range of more than 150m, the Gaussian assumption is likely to hold for up to 40m distances at which the signal strength will be high.

2.3 Energy-efficient Coordination

The second key component of CoCoA is the energy-efficient coordination of the localization algorithm. It is well known that wireless transmissions of data are much more expensive than computation in energy costs. In fact, this is the reason for performing aggressive aggregation of data before forwarding them in wireless sensor networks. Apart from limiting the number of wireless transmissions, another fundamental technique to conserve energy is to wake and sleep periodically. This is useful because typical 802.11 radios (e.g. the Lucent Wavelan – now Orinoco) consume as much energy being idle, as when receiving packets [4]. Thus, significant energy savings are only possible if radios are put in sleep mode instead of idle mode (50mW versus 900mW). However, this requires that nodes be synchronized to some extent. We design CoCoA to require only coarse-grained synchronization achievable through wireless communication among the mobile robots. Once synchronized, the nodes coordinate their wake-sleep periods such that energy is conserved.[1]

The time-line of CoCoA operations is depicted in Figure 2. As shown in the figure, time is divided

[1] The wake-sleep periods regulated by CoCoA can be easily adapted to accommodate scenarios when the radios need to be awake because of the application tasks.

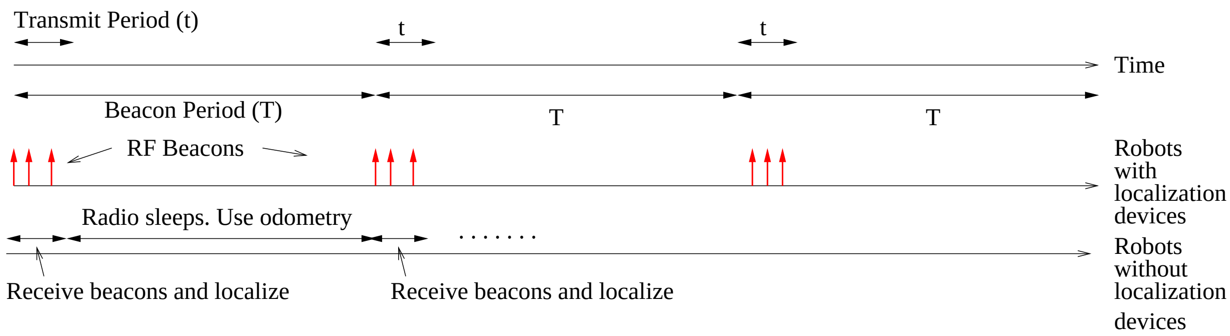


Figure 2. Energy-efficient coordination in CoCoA.

into successive beacon periods (T). Within each beacon period, a small fraction is set aside as a transmit period (t). Both the robots with localization devices and the robots without such devices are coarsely synchronized to follow this time-line. The functions that each type of robot performs are also depicted in Figure 2.

The robots with localization devices transmit k RF beacons during the period t after every beacon period T. k beacons are used for increasing the reliability of beacon delivery. Our evaluation uses k=3. The RF beacon is sent via UDP broadcast. Each beacon, in addition to the IP and UDP headers (20 bytes each), contains the location (x and y coordinates) of the sending robot obtained from the localization devices. The robots without localization devices wake up during each transmit period and use the beacons to execute the localization algorithm presented in the previous section. Following this transmit period, the robots without localization devices use odometry to estimate their position until the next transmit period. At that point the robots throw away their currently estimated positions and find a new position using the beacons. If certain robots do not receive any beacons, they continue with their old estimated position from the previous beacon period.

To achieve coarse-grained synchronization, we use MRMM (Mobile Robot Mesh Multicast), a protocol specifically designed to provide multicast operation in mobile robot networks, proposed in [1]. This allows us to reuse the 802.11 wireless device used for localization to also perform synchronization.

MRMM [1] allows CoCoA to build a virtual mesh of robots from which control packets for synchronization can be delivered to each robot in an energy-efficient manner. MRMM (Mobile Robot Mesh Multicast) is based on the ODMRP (On Demand Multicast Routing Protocol) [5] multicast protocol developed for mobile ad hoc networks. MRMM is an extension of the ODMRP protocol with specific features for efficient operation in mobile robot applications. Both protocols consist of two major phases:

- Mesh construction and maintenance: A mesh is created using a subset of the mobile robots part of the network. The mesh is a structure in the network such that all group members are part of the mesh and certain number of non-members are recruited to forward packets so that no disconnections occur and some redundancy is present. This mesh has to be dynamically reconfigured and adaptable to disconnections due to mobility.

- Data delivery: Data packets are broadcast by robots that are part of the mesh so as to be received by all the group members at that point in time.

MRMM broadcasts a JOIN QUERY message in order to construct a mesh for data delivery. The JOIN QUERY is rebroadcasted by nodes in the mobile robot network. Consider a set of nodes (robots) G that make up the entire mobile robot network. A JOIN QUERY selects from among these G nodes, a set of nodes $F \subseteq G$ that are on the path from the source to all the group members. The set of group members is denoted as M. MRMM exploits the mobility knowledge present in mobile robot networks, i.e. the knowledge of d_{rest}, v and t in order to run a *mesh pruning algorithm*. The objective of the pruning algorithm is to select a new set of nodes $P \subseteq F$ that maximizes the lifetime of the mesh without greatly affecting the redundancy and path lengths of the resulting mesh formed by the set of nodes P. Since P is generally smaller than F, the number of rebroadcasts and consequently the overall control overhead will be reduced in MRMM. Another important consequence of this is that the data packets will travel over a sparser mesh resulting in lower number of data transmissions required to deliver all the data packets. Thus, MRMM will have an improved forwarding efficiency.

CoCoA synchronization using MRMM is depicted in Figure 3 and works as follows. At the beginning of the deployment, one robot designated as the *Sync* robot, begins to send SYNC messages using the data delivery mechanism of MRMM. These SYNC messages are sent at the beginning of every beacon period by the Sync robot. A SYNC message contains the periods T and t. Each node upon receiving a SYNC message, sets up its internal timers for sleeping and waking up the radio according to the specified values of T and t in the packets. This allows a human operator to dynamically adjust these values to depend on the application by notifying the Sync robot to advertise new values. The SYNC message is broadcast down a sparse mesh (depicted in Figure 3), set up dynamically by MRMM, until it reaches every mobile robot. The mesh automatically adjusts to mobility, thus allows CoCoA synchronization to work with mobility as well.

3 Methodology

We use the Glomosim simulator [6] to evaluate the performance of CoCoA. Glomosim is a widely used mobile wireless network simulator with a detailed and accurate physical signal transmission model. We implemented the robot movement model as well as an odometry model (described below) in Glomosim.

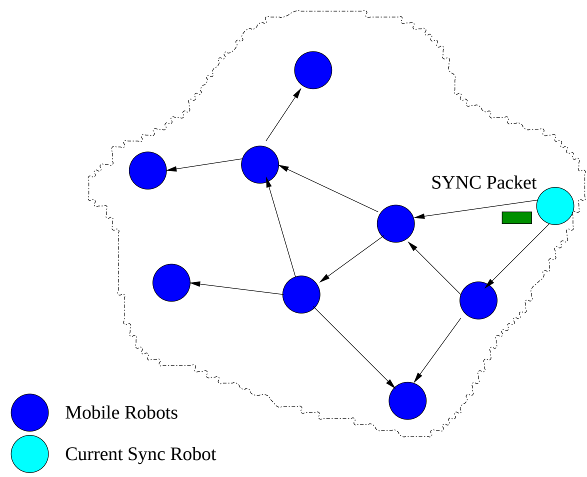


Figure 3. Coarse grained synchronization in CoCoA.

Movement and Odometry Models: The movement model used for the robots is as follows. As the simulation starts, each robot is given a random command to move to a random destination in the given area and starts moving towards the chosen destination with a speed chosen uniformly between 0.1 and v_{max} meters/second. We vary v_{max} as 0.5m/s and 2.0m/s to study its impact on the performance of CoCoA. Once the robot reaches the destination, it is given a new random command (to move to a new destination with a newly chosen speed). This effectively models the movement of a group of robots performing tasks: each robot moves towards a particular area, performs a task, and then moves to the next position.

We implemented an odometry model in Glomosim to simulate the robots actual movement with errors in position estimation. We assume odometry displacement error to be zero-mean Gaussian with standard deviation 0.1m/s and assume the angular odometry error to also be zero-mean Gaussian with standard deviation $10°$.

Energy Model: Since our wireless communication is based on IEEE 802.11, we adopt the energy model and measurements of IEEE 802.11 wireless cards in [4]. We use a wireless network interface with a 2 Mbps bandwidth. Our simulations model the energy spent in transmission, reception, idling and sleeping along with the energy spent in powering the card on and off.

Metrics: The following metrics are evaluated for the CoCoA system: (1) *Localization error* – The distance between the real position of the robot and where the robot estimates itself to be; and (2) *Energy consumption* – The total amount of energy consumed by the the robot team for communication and localization. This includes energy spent during sending and receiving both data and control packets as well as energy spent when the wireless device is idle or in sleep mode.

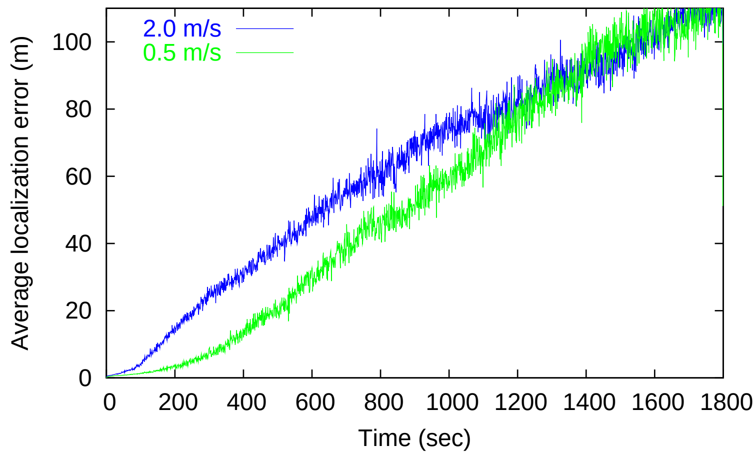


Figure 4. Localization error over time using only odometry.

4 Performance Evaluation

In this section, we evaluate the performance of CoCoA and investigate the impact of the different configuration parameters. In all the simulation experiments, we assume a total of 50 mobile robots operating in an area of size of $40000m^2$. Half of the robots are equipped with localization devices unless otherwise stated. The robots move according to the movement model defined above. The simulation time is 30 minutes.

The outline of the evaluation is as follows: We first examine how well the extreme approach of only using odometry given an initial position estimate performs. We then evaluate the performance of our RF localization algorithm by itself. Next, we evaluate the performance of CoCoA, which combines odometry with RF localization in comparison to the previous two approaches. Finally, we study the impact of various parameters on CoCoA performance and demonstrate its energy-efficiency.

4.1 Localization using only odometry

In this section, we show how the localization error grows over time when the mobile robots rely only on odometry. In this experiment, the robots are provided with their initial coordinates, but during the simulation duration of 30 minutes, they only use odometry to maintain an estimate of their positions. Figure 4 depicts the average localization error over all the 50 robots at each second for the simulation duration, for two different maximum speeds: 0.5m/s and 2m/s.

Figure 4 shows that, in spite of the instantaneous variations, the average localization error increases significantly over time. After half an hour, it becomes larger than 100m for both speeds. Thus, using only odometry is not accurate enough for a robot to maintain an accurate estimate of its position over a long period of time, even when the robot is initially provided with its real position. The reason for

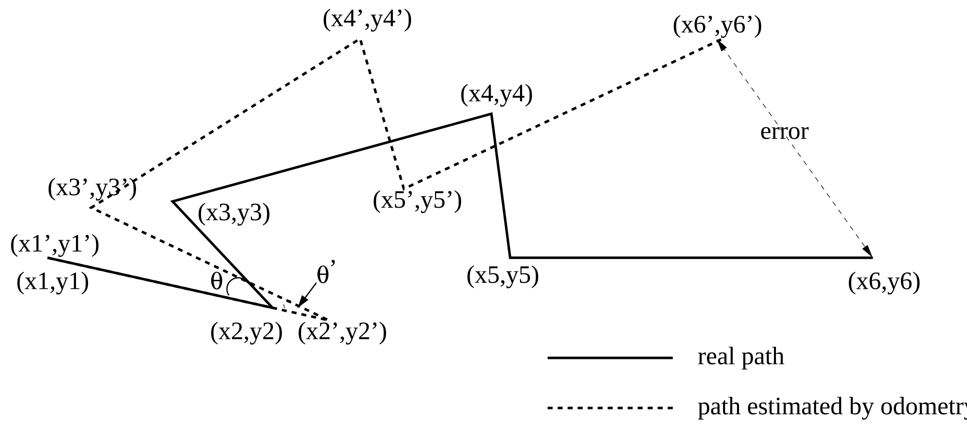


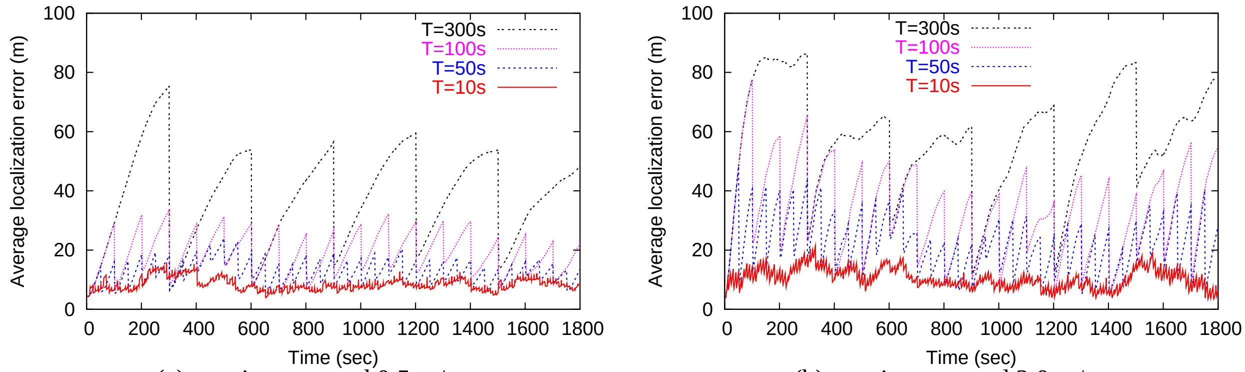
Figure 5. An example of odometry error.

this is that the odometry error is accumulated over time. Figure 5 explains it with an example. This example shows the path followed by a robot and the estimated path using odometry. We denote by (x_i, y_i) the real positions of the robot and by (x'_i, y'_i) the positions estimated using odometry. As we can see in this figure, the robot starts at position (x_1, y_1). At that moment the position estimate (x'_1, y'_1) is the same as the real position (we assume that the robot is provided with its initial position). When the robot reaches at position (x_2, y_2), due to the odometry displacement error, it estimates itself to be at (x'_2, y'_2), which is different than (x_2, y_2). Furthermore, when the robot turns by θ at (x_2, y_2), due to the angular odometry error, it estimates a turn by θ', which is different than θ. This, combined with the displacement error will result in the robot estimating itself to be at (x'_3, y'_3), when it actually is at (x_3, y_3). In general, if we use only odometry for the location estimation, the error can be very large after a long period of time. For example, in Figure 5, the robot's final position estimate (x'_6, y'_6) is very far from its real final position (x_6, y_6).

This shows that odometry is useful only for a small period of time, after which, robots need to update their position estimates using external devices. Figure 4 helps us to define the period T during which the robots can use odometry without a significant degradation in their position estimates. Every T seconds, the robots equipped with localization devices have to transmit beacons with their coordinates, and the other beacons reset their position estimates. Obviously, the period T depends on the accuracy required for a specific application.

4.2 RF Localization

The previous section demonstrated the need for external devices to improve localization accuracy. In this section, we use the Bayesian RF localization method described in Section 2.2 *without odometry* and measure the localization accuracy achieved. For this experiment, robots with localization devices every T seconds transmit beacons with their coordinates for a transmit period t = 3 seconds, and in the



(a) maximum speed 0.5 m/s

(b) maximum speed 2.0 m/s

Figure 6. Localization error over time using only RF localization. Impact of different beacon periods is depicted.

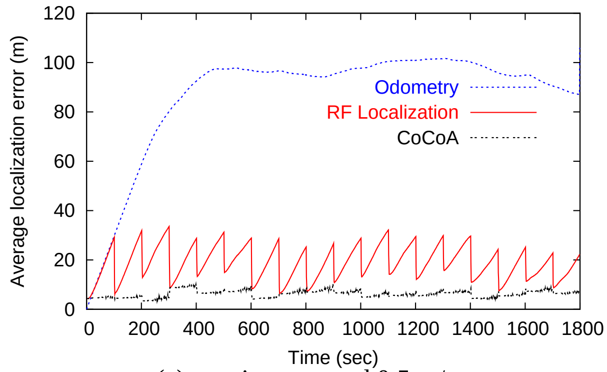
same time, the rest of the robots execute the localization algorithm and update their position estimates, which remain the same, until the T-second period expires.

The results of our evaluation are depicted in Figure 6 for different values of T. Note that the localization error is only reported for nodes that are not equipped with localization devices. The results show that compared to using odometry, the localization accuracy is significantly improved by using RF localization. We observe that the localization error becomes minimum in the beginning of every period T, and it increases with time, as the position estimates become stale. Figure 6 also shows that the accuracy depends on the selected period T. Larger T reduces the accuracy over time, because the robots update their positions less often. On the other hand, a larger T can reduce energy consumption as we will demonstrate in the later half of the evaluation.

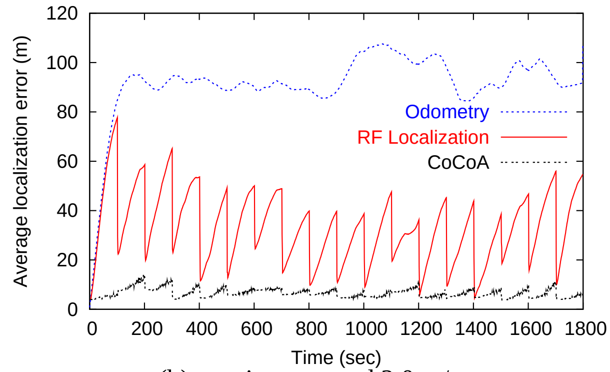
In addition to providing improved accuracy in comparison to odometry, RF localization does not require an initial position which makes it easy for robots to be deployed arbitrarily. However, in CoCoA we propose to incorporate odometry information as well as RF localization so that the localization error is reduced when beacons are not received (e.g. due to network partitions or in the rest of a beacon period, before the robots with localization devices transmit again their coordinates). We explore the performance of CoCoA in the next section.

4.3 CoCoA: RF Localization and Odometry

In this section, we evaluate the performance of CoCoA, which combines RF localization and odometry. We start by comparing the localization error over time for CoCoA in the previous scenario, against the two previous approaches: using only odometry and using only the RF localization algorithm. For this comparison we selected T = 100 sec. The evaluation results are shown in Figures 7(a), 7(b) for the



(a) maximum speed 0.5 m/s



(b) maximum speed 2.0 m/s

Figure 7. Localization error over time for T = 100 seconds using (i) only odometry, (ii) only the RF localization algorithm and (iii) a combination of them (CoCoA).

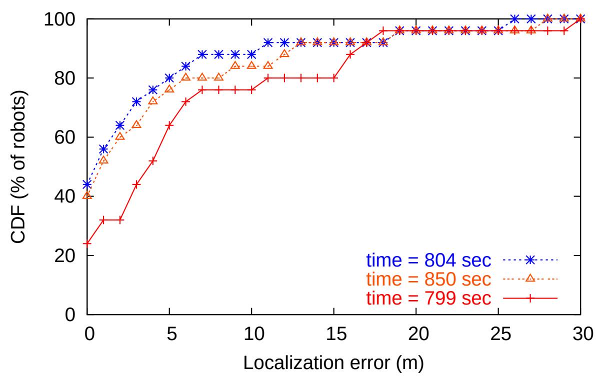


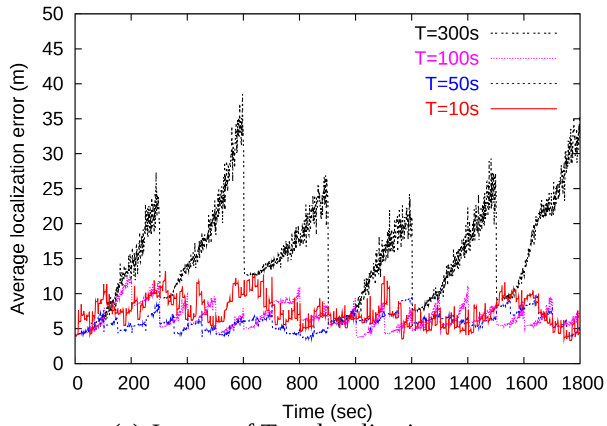
Figure 8. CDF for the localization error at three time instances.

two different maximum speeds.

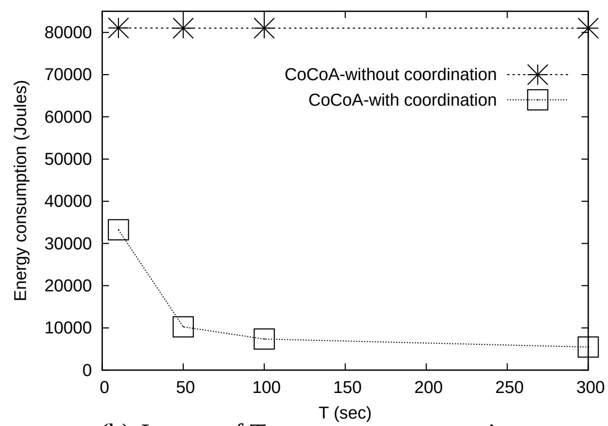
We observe that CoCoA offers significantly higher accuracy, compared to the other two approaches, since it combines the advantages of both. For example, the average localization error over time for a maximum speed 2 m/s is only 6.5m with CoCoA, while it is about 33m when only the localization algorithm is used.

Figure 8 shows the Cumulative Distribution Function (CDF) of the localization error in CoCoA for three different time instances (in the end of a beacon period just before the start of the next transmit period, in the end of a transmit period after localization has been completed, and in the middle of a beacon period while the radio is sleeping, $\frac{T}{2}$ seconds after the transmit period is over).

As expected, the localization is best right after beacons are received at time 804s. Since T=100s, the time instant 804s is right after a transmit period has occurred. The CDF also shows that the locations do deteriorate over time but not significantly. These results demonstrate that CoCoA improves not only the overall average localization error but also reduces the localization error of a large fraction of the



(a) Impact of T on localization error



(b) Impact of T on energy consumption

Figure 9. Impact of beacon period (T) on CoCoA localization error and energy consumption. 50% of the nodes are equipped with localization devices.

nodes. For example, Figure 8 shows that more 90% of the robots have a localization error lower than 10m.

In the next two subsections, we study the performance of CoCoA in more detail, in relation with two important parameters: the period T and the number of robots equipped with localization devices. CoCoA offers one more important advantage, in addition to the localization accuracy. It reduces the energy consumption, by putting the radios of the robots to sleep mode during each period T and waking them up in the beginning of the next period. We also study this effect in the next section.

4.3.1 Impact of Beacon Period

In this section, we evaluate the impact of the beacon period T and discuss how this should be chosen to provide a good tradeoff between localization accuracy and energy consumption. The evaluation results are shown in Figures 9(a), 9(b).

Figure 9(a) shows the localization error over time for four different beacon periods: 10, 50, 100 and 300 sec. In general, a small T improves the localization accuracy over time, since it gives the robots the chance to update frequently their position estimates. However, we surprisingly observe that for very small T (T = 10 sec) the average error over time becomes worse (it is about 7m for T = 10 sec, 5m for T = 50 sec and 6.6m for T = 100 sec). This shows that the position updates due to the localization algorithm are not always accurate. Some “bad” beacons, e.g. beacons received from long distances may deteriorate the performance of the algorithm. For this reason, the algorithm should not be executed arbitrarily frequently, but only when the error due to odometry has become quite large.

On the other hand, a very large value for T reduces energy consumption, since the radios are in sleep mode most of the time, and there is a limited number of transmissions/receptions which consume much energy. This is shown in Figure 9(b). In this figure we also show the energy consumption for CoCoA without coordination, i.e., when the radios do not go to the sleep mode, but they remain idle when they do not send or receive beacons. We observe that in that case the energy consumption is 2.6 to 8 times larger than the case with coordination, depending on the beacon period T.

From Figure 9(a) we observe that the localization error does not change a lot when T changes from 50 sec to 100 sec. Also from Figure 9(b) we observe that the energy consumption changes very slowly for T larger than 50 sec. Hence, the values between 50 and 100 sec are the best for the beacon period T, offering both high accuracy and low energy consumption.

4.3.2 Impact of Number of Localization Devices

In this section, we evaluate the performance of CoCoA with different fractions of nodes equipped with localization devices. Reducing the number of such nodes reduces costs but can lead to inaccuracies in localization due to fewer beacons in the system. A small number of robots equipped with localization devices may affect the localization accuracy in two ways. First, some robots may not be able to receive any beacons, and they will localize themselves using only odometry, for one or more beacon periods T. Second, a robot may receive very few beacons, all from long distances or from one side of the robot, and in that case the localization algorithm will have poor accuracy.

The evaluation results are shown in Figure 10. For this experiment we varied the number of robots with localization devices from 5 to 35. In this figure we observe that the average error over time does not increase a lot, when we reduce the robots with localization devices from 35 (5.2m) to 25 (5.9m). Thus, by installing localization devices on only half of the robots, we can achieve an average localization error over time less than 6m and a maximum error less than 10m, by significantly reducing the cost. Depending on the application, we can reduce the cost even more, by using only 15 robots with localization devices, which gives an average error over time of about 8m and maximum error less than 12m.

In summary, the results demonstrate that CoCoA is an energy-efficient, cost-effective and easily deployable architecture for localization of mobile robots in infrastructure-less environments.

5 Related Work

Robot localization is a widely studied problem in robotics. It can be defined as the process of maintaining an estimate of the robot's location with respect to its environment. We can distinguish two cases for localization. The case when there is no a priori knowledge of the robot's location is known as *global localization*, and it is a particularly challenging task. The case when the robot has

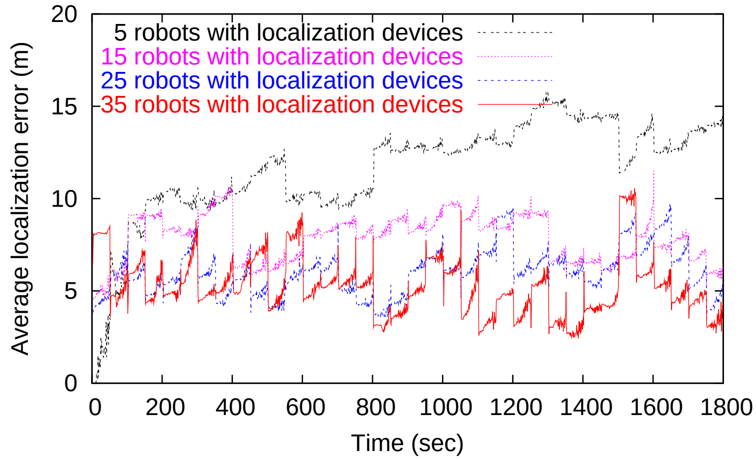


Figure 10. Impact of number of robots with localization devices on localization error.

some initial estimate about its position, and it tries to refine this estimate, using observations from the environment, is known as *pose maintenance*. Many different localization techniques have been proposed since the problem first appeared. Some of them are dead reckoning, triangulation, Kalman Filter, Bayesian approaches and topological approaches.

Dead reckoning is the simplest approach for pose maintenance. The robot keeps track of how far it moves in each direction, and then it adds these distances to its initial position estimate, in order to obtain an updated position. The most common way to do this is by using an odometer. The use of an odometer only is not enough for accurate localization, because the odometry errors are added to the initial position estimate, and the total error is accumulated over time. Hence, for long-term localization, the robot has to periodically update its position using external references, e.g., landmarks equipped with GPS devices [7, 8, 9, 10, 11, 12, 13, 14, 15, 16].

When distance to three or more landmarks is known, triangulation or multilateration can be used respectively to estimate the robot's position. This approach depends highly on the quality of the distance measurements to the known landmarks. If the measurements are not accurate enough, which is usually the case for RF signals, due to phenomena such as fading, shadowing, multipath propagation, etc., the localization error can be large. In such cases, more expensive devices have to be used for measurements (e.g. acoustic devices and lasers) or more complicated techniques, such as Kalman Filter, Bayesian or topological approaches.

Kalman Filter [8, 9, 10, 11] represents the robot's position estimate as a Gaussian distribution, and it uses sensor data from both dead reckoning (odometry) and external observations to obtain a new distribution. Kalman Filter works only when we can assume that all distributions are linear. If this assumption does not hold, Bayesian approaches can be used instead, such as Markov localization [12, 13], or Monte Carlo localization [14, 15]. All Bayesian approaches use a grid representation of the environment, and estimate the probability of the robot to be in each grid. Another way of representing the environment is by using a topological map, represented as a generalized Voronoi graph [16].

In [17], the authors propose the use of 802.11 wireless devices for robot localization and apply a method similar to ours – system calibration followed by Bayesian interference. The main difference of this work compared to CoCoA is that it was proposed for indoor localization. This means that signal strength measurements cannot be mapped to distances using Gaussian functions, since in indoor environments signal strength is highly affected by phenomena such as multipath propagation and fading. Instead, a signal strength map is built during system calibration and the system learns the signal strength observed at a number of positions. Then, Bayesian interference uses the measured signal strength value (and not the distance), and selects among the learned positions the one with signal strength closest to the measured value as the robot's position.

Multi-robot localization, where robots cooperate to localize each other, has also been studied during the last 10 years. The basic principles can be found in [7], where the authors propose a scheme called "Cooperative Positioning". In this scheme some of the robots act as landmarks, helping others to estimate their positions. No localization devices are used, hence robots periodically change roles; localized robots stop moving and act as landmarks, while robots acting as landmarks are allowed to move and estimate their new positions, based on the new landmarks. Obviously this adds accumulated errors. To minimize these errors extra constraints are imposed on robots movement.

Different approaches can be found in [18], [19], [20], and [21]. In [18] a sample-based version of Markov localization is proposed. Teams of robots localize themselves in the same environment using external landmarks, and probabilistic methods are then employed to synchronize each robot's belief whenever a robot detects another. The algorithm requires that the robots be equipped with cameras and laser range-finders in order to detect each other.

A Kalman-Filter approach is used in *Collective Localization* proposed in [19]. Each robot collects information about its own motion and shares this information with the rest of the team during update cycles. A Kalman Filter processes the available positioning information from all the members and produces a pose estimate for all of them. The paper focuses on how to decompose the single Kalman filter in smaller communicating filters. Note that this method relies entirely on external landmarks; no attempt is made to sense other robots. In [22], the method is extended to include resource-constrained localization. Constraints on the available bandwidth, as well as communication and processing requirements limit the number of measurements that can be processed at each time step. The authors take these constraints into account and formulate an optimization problem, whose solution provides the optimal sensing frequencies in order to maximize the position accuracy for the whole group.

A different approach in which no external landmarks are required can be found in [20]. In this method each robot can measure the relative pose of nearby robots (using either vision or scanning laser range-finders) and changes to its own pose (using either odometry or inertial measurement units). A combination of Maximum Likelihood Estimation and numerical optimization is then used to infer the relative pose of each robot in the team.

Finally in [21] a method for simultaneous localization and map building (SLAM) is proposed. The robots estimate the position of static landmarks and localize themselves with respect to landmarks and other robots, using a set theoretic approach.

In contrast to these previous works, CoCoA is a general purpose architecture which allows mobile robots to cooperate and coordinate in localizing each other, without the need of external landmarks, and without the need of expensive localization devices installed on all robots. Moreover, CoCoA is not tied to a specific localization technique. In this paper, we have implemented a Bayesian technique in the CoCoA localization component. Other approaches could be integrated in CoCoA as well. CoCoA provides the means for any specific localization technique to be used in a cooperative and coordinated manner.

6 Conclusion and Future Work

In this paper, we proposed CoCoA: an architecture for low-cost, quickly-deployable and energy-efficient localization for mobile robot teams in infrastructure-less environments. CoCoA equips few robots with localization devices, such that in the process of performing their own tasks, they cooperate and help to localize the other robots. In addition, CoCoA coordinates this localization process using multicast to put wireless devices in sleep mode and provides significant energy savings. CoCoA provides reasonably accurate locations which can be useful for many applications. For example, CoCoA coordinates are good enough to enable scalable geographic routing [23] of messages and data among the robots or to a controller. The average localization error is about 8m when only one third of the robots are equipped with localization devices. This can be useful for many applications, such as search and rescue operations, since survivors can be located within 8m. Pinpointing the exact location of the survivor is then trivial once more resources are deployed to the area.

There are also several interesting avenues for further investigation. One area is to use the robots that do not have localization devices but are already localized to also initiate beaconing. This could potentially reduce the need for robots equipped with localization devices and lower costs. On the other hand, it is hard to ascertain the goodness of the location a particular node has and using such techniques could potentially increase localization errors. Another avenue for investigation is whether and how CoCoA coordinates can be used to improve the performance or speed of more fine-grained localization techniques (e.g. using a laser ranger). We are also interested in determining how transmission power control can be used to increase the distance that nodes in the CoCoA architecture can cooperate. It is interesting to investigate the noise distributions of RF beacons when operating over special hardware that supports power control. In this paper we evaluated CoCoA using simulations. A testbed evaluation requires a large number of robots, which are not currently available in our testbed. This is a focus of our future work. However, we believe CoCoA would work well in a real world deployment since our simulation model is based on data from real wireless signal measurement tests and not a theoretical model. Thus, we expect a close correlation between our simulation results and a real world deployment of CoCoA.

Acknowledgment

This work was supported in part by the National Science Foundation under Grant IIS-0329061.

References

[1] S. M. Das, Y. C. Hu, C. Lee, and Y.-H. Lu, "An efficient group communication protocol for mobile robots," in *Proc. of IEEE ICRA*, 2005.

[2] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem." *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, 2001.

[3] M. L. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," in *Proc. of MASS*, September 2004.

[4] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. of IEEE INFOCOM*, April 2001.

[5] S.-J. Lee, M. Gerla, and C.-C. Chiang, "On-Demand Multicast Routing Protocol," in *Proc. of IEEE WCNC*, September 1999.

[6] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for parallel simulation of large-scale wireless networks," in *Proc. of PADS Workshop*, May 1998.

[7] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *Proc. of the IEEE International Conference in Robotics and Automation*, 1994.

[8] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Journal of Robotic Reasearch*, vol. 5(4), pp. 56–68, Winter 1987.

[9] A. Davison and N. Kita, “3d simultaneous localisation and map-building using active vision for a robot moving on undulating terrain,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, December 2001.

[10] J. Guivant and E. Nebot, “Optimization of the simultaneous localization and map building algorithm for real time implementation,” *Journal of Robotic Reasearch*, vol. 17(10), pp. 565–583, 2000.

[11] J. F. Leonard and H. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” *IEEE Transactions Robotic and Automation*, vol. 7(3), pp. 376–382, 1991.

[12] D. Fox, W. Burgard, and S. Thrun, “Markov localization for mobile robots in dynamic environments,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.

[13] K. Konolige and K. Chou, “Markov localization using correlation,” in *Proc. of IJCAI*, 1999.

[14] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” in *Proc. of AAAI*, July 1999.

[15] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, “Robust monte carlo localization for mobile robots,” *Artificial Intelligence Journal*, 2001.

[16] B. Kuipers and Y.-T. Byun, “A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations,” *Journal on Robotics and Automatic Systems*, vol. 8, pp. 47–63, 1991.

[17] A. Ladd, K. Bekris, A. Rudys, L. Kavraki, and D. Wallach, “On the feasibility of using wireless ethernet for indoor localization,” *IEEE Transactions on Robotics and Automation*, vol. 20, 2004.

[18] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, “A probabilistic approach to collaborative multirobot localization,” *Autonomous Robots*, vol. 8, 2000.

[19] S. Roumeliotis and G. Bekey, “Distributed multirobot localization,” *IEEE Transactions on Robotics and Automation*, vol. 18, 2002.

[20] A. Howard, M. Mataric, and G. Sukhatme, “Localization for mobile robot teams using maximum likelihood estimation,” in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2002.

[21] M. D. Marco, A. Garulli, A. Giannitrapani, and A. Vicino, “Simultaneous localization and map building for a team of cooperating robots: a set membership approach,” *IEEE Transactions on Robotics and Automation*, vol. 19, 2003.

[22] A. Mourikis and S. Roumeliotis, “Optimal sensing strategies for mobile robot formations: Resource-constrained localization,” in *Proc. of Robotics: Science and Systems*, 2005.

[23] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” in *Proc. of ACM DialM Workshop*, August 1999.