

Cooperative Distributed Sensors for Mobile Robot Localization

Zhiwei Liang, Songhao Zhu

College of Automation, Nanjing University of Posts and Telecommunications, Nanjing, China

E-mail: lzhu_ly@hotmail.com

Received December 24, 2009; revised February 22, 2010; accepted March 17, 2010

Abstract

This paper presents a probabilistic algorithm to collaborate distributed sensors for mobile robot localization. It uses a sample-based version of Markov localization—Monte Carlo Localization (MCL), capable of localizing mobile robots in an any-time fashion. During robot localization given a known environment model, MCL method is employed to update robot's belief whichever information (positive or negative) attained from environmental sensors. Meanwhile, an implementation is presented that uses color environmental cameras for robot detection. All the parameters of each environmental camera are unknown in advance and need be calibrated independently by robot. Once calibrated, the positive and negative detection models can be built up according to the parameters of environmental cameras. A further experiment, obtained with the real robot in an indoor office environment, illustrates it has drastic improvement in global localization speed and accuracy using our algorithm.

Keywords: Monte Carlo Localization, Environmental Camera, Positive Information, Negative Information

1. Introduction

Mobile robot localization is the problem of estimating a robot's pose (location, orientation) relative to its environment. The localization problem is a key problem in mobile robotics. There are two classes of localization problem, position tracking and global localization. In position tracking, a robot knows its initial position [1] and only needs to reduce uncertainty in the odometer reading. If the initial position is not known or the robot is kidnapped to somewhere, the problem is one of global localization, *i.e.*, the mobile robot has to estimate its global position through a sequence of sensing actions [2]. In recent years, a flurry of publications on localization document the importance of the problem. Occasionally, it has been referred to as "the most fundamental problem to providing a mobile robot with autonomous capabilities" [3].

So far, virtually all existing work addresses localization only using sensors onboard mobile robot. However, in robot navigation, the robot cannot always determine its unique situation only by local sensing information since the sensors are prone to errors and a slight change of the robot's situation deteriorates the sensing results. Along with the rapid development of computer networks

and multimedia technology, research on how to make an 'intelligent' environment for the robot to fulfill the same functions makes sense, especially in home environment. In this case, various sensors are embedded into the environment (environmental sensors), and communication between the robot and environmental sensors is utilized. Sogo *et al.* [4] proposed a distributed vision system for navigating mobile robots in a real world setting. To obtain robustness and flexibility, the system consisted of redundant vision agents connected to a computer network. These agents provided information for robots by organizing the communication between vision agents. Morioka *et al.* [5] defined the space in which many vision sensors and intelligent devices are distributed as an intelligent space. Mobile robots exist in this space as physical agents that provide humans with services. A concept called a distributed modular robot system was also proposed in [6]. In that robot system, a modular robot was defined as a mono-functional robot (either a sensor or an actuator) with a radio communication unit and a processing unit. Such robots were usually small and could be easily attached to operational objects or dispersed into the environment. A modular robot system for object transportation was developed by using several distributed-placed camera modules and wheel modules. All studies mentioned above mostly focus on the struc-

ture of system, while don't put forward an effective method to incorporate the information of environmental sensors. On the other hand, they only apply the positive information (it represents a sensor detects the robot) to localize robot, while don't take into account how to make use of the negative information which represents that a sensor doesn't detect the robot.

The aim of this paper is to show how positive and negative information of environmental sensors can be incorporated in robot localization. Therefore, an efficient probabilistic approach based on *Markov Localization* [7-9] is proposed. In contrast to previous research, which relied on grid-based or coarse-grained topological representations of the robot's state space, our approach adopts a sampling-based representation [10]—Monte Carlo Localization (MCL), which is capable of approximating a wide range of belief functions in real-time. Using the positive and negative detection model of environmental sensors, the MCL algorithm can improve localization accuracy and shorten the localization time. In terms of practical applications, while our approach is applicable to any sensor capable of detecting robot, we present an implementation that uses color environmental cameras for robot localization. The location and parameters of all environmental cameras are unknown and need to be calibrated by robot. Once getting the cameras' parameters, the positive and negative detection models can be attained. Experimental results, carried out with two environmental cameras fixed in an indoor environment, illustrate the appropriateness of the approach in robot global localization.

This paper is organized as follows. In Section 2, the MCL only depending on the robot's own sensor is introduced. Section 3 extends the algorithm to integrate the positive and negative information coming from environmental sensors. Experimental results are shown in Section 4. Finally, in Section 5, our conclusions are derived.

2. Monte Carlo Localization

In this section we will introduce our sampling-based localization approach only depending on robot itself. It is based on Markov localization [7-9], which provides a general framework for estimating the position of a mobile robot. Markov localization maintains a belief $Bel(L^{(r)})$ over the complete three-dimensional state space of the robot. Here, $L^{(r)}$ denotes a random variable and $Bel(L^{(r)}=l)$ denotes the robot's belief of being at location l , representing its x - y coordinates (in some Cartesian coordinate system) and its heading direction θ . The belief over the state space is updated whenever

the robot moves and senses.

Monte Carlo localization relies on sample-based representations for the robot's belief and sampling/importance resampling algorithm for belief propagation [11,12]. The sampling/importance resampling algorithm has been introduced for Bayesian filtering of nonlinear, non-Gaussian dynamic models. It is known alternatively as the bootstrap filter [13], the Monte-Carlo filter [10], the Condensation algorithm [14], or the survival of the fittest algorithm [15]. All these methods are generically known as *particle filters*, and a discussion of their properties can be found in [16].

More specifically, MCL represents the posterior beliefs $Bel(L)$ over the robot's state space by a set of N weighted random samples denoted $S = \{s_i | i = 1 \dots N\}$. A sample set constitutes a discrete distribution. However, under appropriate assumptions (which happen to be fulfilled in MCL), such distributions smoothly approximate the correct one at a rate of $1/\sqrt{N}$ as N goes to infinity. Samples in MCL are of the type $\langle l, p \rangle$, where l denotes a robot position in x - y - θ space, and $p \geq 0$ is a numerical weighting factor, analogous to a discrete probability. For consistency, we assume $\sum_{i=1}^N p_i = 1$.

In analogy with the general Markov localization approach, MCL propagates the belief as follows:

1) Robot motion. When a robot moves, MCL generates N new samples that approximate the robot's position after a motion measurement a . Each sample is generated by randomly drawing a sample from the previously computed sample set, with likelihood determined by their p -value. Let l' denote the $\langle x, y, \theta \rangle$ position of this sample. The new sample's l is then determined by generating a single, random sample from the distribution $P(l|l', a)$, using the observed motion a . The p -value of the new sample is N^{-1} . Here $P(l|l', a)$ is called the motion model of the robot. It models the uncertainty in robot motion.

2) Environment measurements are incorporated by re-weighting the sample set, which is analogous to the application of Bayes rule to the belief state using importance sampling. More specifically, let $\langle l, p \rangle$ be a sample. Then

$$p \leftarrow \alpha P(o|l) \quad (1)$$

where o is a sensor measurement, and α is normalization constant that enforces $\sum_{i=1}^N p_i = 1$. $P(o|l)$, also called the environment perceiving o given that the robot is at position l . The incorporation of sensor readings is typically performed in two phases, one in which

p is multiplied by $P(o|l)$, and one in which the various p -value are normalized.

For proximity sensors such as laser range-finders which is adopted in my approach, the probability $P(o|l)$ can be approximated by $P(o|o_l)$, which is the probability of observing o conditioned on the expected measurement o_l at location l . The expected measurement, a distance in this case, is easily computed from the map of the environment using ray tracing. **Figure 1** shows a perception model for laser sensor. Here the x -axis is the distance o_l expected given the world model, and the y -axis is distance o measured by the sensor. The function is mixture of a Gaussian (centered on the correct distance o_l), a Geometric distribution (modeling overly short readings) and a Dirac distribution (modeling max-range readings). It integrates the accuracy of the sensor with the likelihood of receiving a “random” measurement (e.g., due to obstacles not modeled in the map [9]).

3. Cooperative Distributed Sensor Localization

In this section, we will first describe the basic statistical mechanism for cooperative distributed sensor localization and then its implementation using MCL. The key idea of cooperative distributed sensor localization is to integrate measurements taken at different platforms, so that robot can benefit from information gathered by environmental sensors, which are embedded in the environment, other than itself. The information coming from environmental sensors includes “positive” detections, *i.e.*, cases where an environmental sensor does detect the robot, and “negative” detection events, *i.e.*, cases where an environmental sensor does not see the robot.

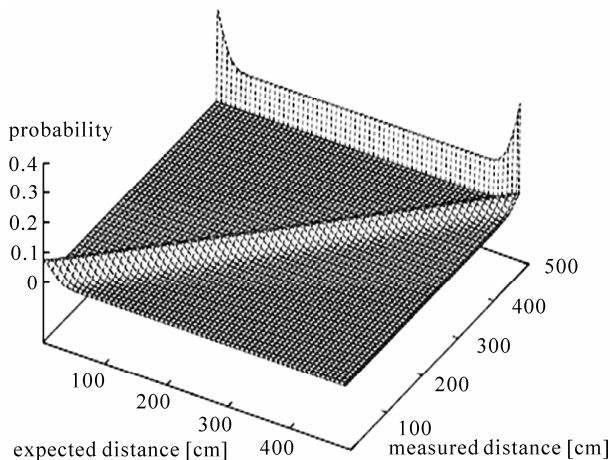


Figure 1. Perception model for laser range finders.

3.1. Positive Detections

When one environmental sensor detects the robot, sample set is updated using the detection model, according to the update equation

$$Bel^{(r)}(l) \leftarrow Bel^{(r)}(l) \sum_{l'} P(L^{(r)} = l | L^{(m)} = l', r_+^{(m)}) Bel^{(m)}(l') \quad (2)$$

Notice that this equation requires the multiplication of two densities. The crucial component is the probabilistic detection mode $P(L^{(r)} = l | L^{(m)} = l', r_+^{(m)})$ which describes the conditional probability that robot is at location l , given that sensor m is at possible location l' and perceives robot with positive measurement $r_+^{(m)}$. From a mathematical point of view, our approach is sufficiently general to accommodate a wide range of sensors for robot detection, assuming that the conditional density $P(L^{(r)} | L^{(m)}, r_+^{(m)})$ is adjusted accordingly. However, for environmental camera it is not necessary to build the probabilistic detection mode $P(L^{(r)} = l | L^{(m)} = l', r_+^{(m)})$ and the environmental camera localization model $Bel^{(m)}(l')$ respectively. As a substitute, joint detection model

$$P(L^{(r)} = l | r_+^{(m)}) \triangleq \sum_{l'} P(L^{(r)} = l | L^{(m)} = l', r_+^{(m)}) Bel^{(m)}(l') \quad (3)$$

is constructed directly according to the environmental cameras' parameters. Before integrating positive information of environmental cameras into robot's belief, the cameras' parameters need to be calibrated by robot.

3.1.1. Camera Self-Calibration

In our method, all parameters of the environmental cameras are unknown in advance and their visual fields are not overlaid each other. So, in order to apply them to localize the robot, each camera's parameters need to be calibrated at first. Assuming that the system is always ready for using in different environments, calibration instruments (such as patterns and measuring devices) may more or less hinder portability. Our objective is to introduce a self-calibration concept [17] into the system and take the mobile robot as a calibration instrument. Because the visual fields of all cameras are not overlaid, each camera's calibration is independent.

During the calibration, the robot location is known. When the robot moves depending on laser and odometry in visual field of any environmental camera, the camera does detect the robot and gathers the relative data between the robot global location and detected image pix-

els. The sample space of relative data is designed to satisfy a condition that the distance between two neighbor global locations of relative data is more than 0.2 m. Once the number of relative data sums up to a threshold which is set as 200 in this paper, camera calibration can be conducted. Because the mobile robot always moves in a plane, the coplanar camera calibration method of Tsai is adopted here [18].

In addition, unlike ordinary calibration devices, the mobile robot is much less accurate when moving. As the most distinct point of the robot's error, it is cumulative and increase over time or repeated measurements. Moreover, the random motion input of the robot, which may take too much time, is not suitable for our method. For all these reasons, robot's motion during calibration process should be designed to avoid serious calibration

error and to meet the accuracy demands of calibration. In our method, the robot in the cameras' visual field moves as a zig, which is shown in **Figure 2**.

3.1.2. Detection

To determine the location of the robot, our approach combines visual information obtained from environmental cameras. Camera images are used to detect mobile robot and determine the position of the detected robot. The two rows in **Figure 2** shows examples of camera images recorded in a room. Each image shows a robot, marked by a unique, colored marker to facilitate its recognition. Even though the robot is only shown with a fixed orientation in this figure, the marker can be detected regardless of the robot's orientation.

To find the robot in a camera image, our approach first

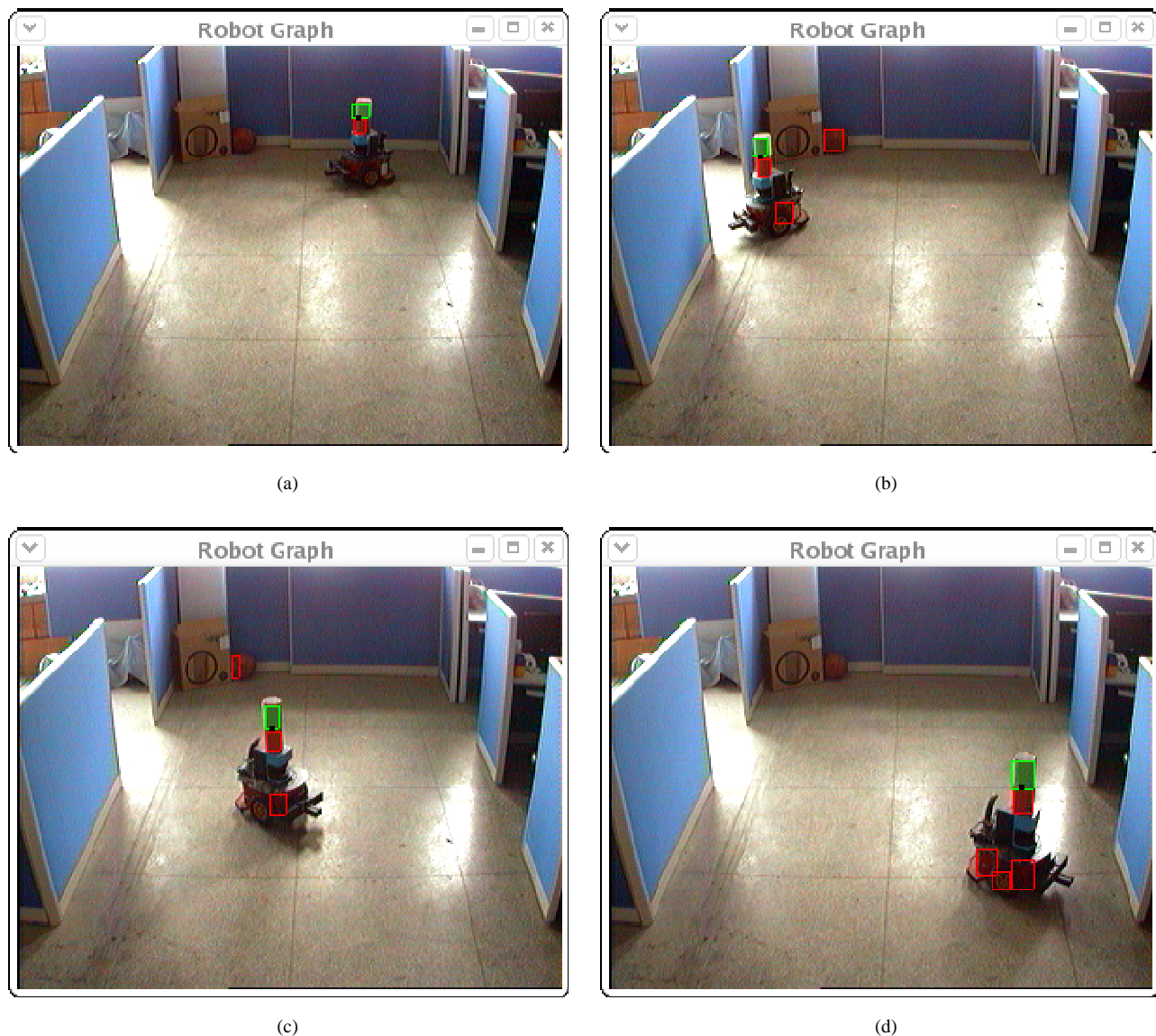


Figure 2. Image sequences of successful detecting the robot which moves as a zig.

filters the image by employing local color histograms and decision trees tuned to the colors of the marker. Threshold is then employed to search for the marker's characteristic color transition. If found, this implies that the robot is present in the image. The small black points, superimposed on each marker in the images in **Figure 2**, illustrate the center of the marker as identified by this distributed environmental camera.

Once a robot has been detected, the current environmental camera is analyzed for the location of the robot in image coordinates. Then transform the detection pixels in image coordinates to positions in world coordinates according to the calibrated parameters of the camera. Here, tight synchronization of photometric data is very important, especially because the mobile robot might shift and rotate simultaneously when it is sensed. In our framework, sensor synchronization is fully controllable because all data is tagged with timestamps.

3.1.3. Joint Detection Model

Next, we have to devise a joint detection model of the type $P(L^{(r)} = l | r_+^{(m)})$. To recap, $r_+^{(m)}$ denotes a positive detection event by the m -th environmental camera, which comprises planar location of the detected robot in world coordinates. The variable $L^{(r)}$ describes the location of the detected robot. As described above, we will restrict our considerations to "positive" detections, *i.e.*, cases where an environmental camera did detect a robot.

The joint detection model is trained using data. More specifically, during training we assume that the exact location of robot is known. Whenever an environmental camera takes an image which is analyzed as to whether robot is in its visual field, it is to exploit the fact that the locations of robot are known during training. Then, the image is analyzed, and for detected robot global location is computed according to the calibrated parameters of the environmental camera above. This data is sufficient to train the joint detection model

$$P(L^{(r)} = l | r_+^{(m)}) = (2\pi)^{-\frac{1}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (Al - r_+^{(m)})' \Sigma^{-1} (Al - r_+^{(m)}) \right\} \quad (4)$$

where $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$, $\Sigma = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$, σ_x^2 and σ_y^2 represent mean square error in x direction and y direction respectively. Let $r_+^{(m)}$ as coordinate origin, and the Gaussian model showed in **Figure 3** models the error in the estimation of robot's location. Here the x -axis represents the error of x direction in the world

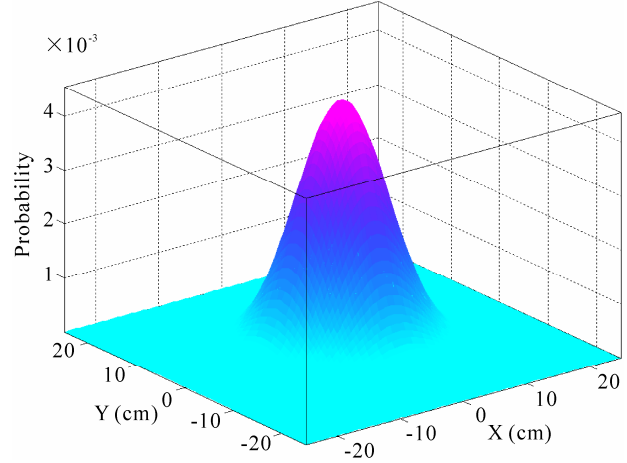


Figure 3. Gaussian model representing camera's detection error.

coordinates, and the y -axis the y direction error. The parameters of this Gaussian model has been obtained through maximum likelihood estimation [19] based on the training data. As is easy to be seen, the Gaussian model is zero-centered along both dimensions, and it assigns low likelihood to large errors. Assuming independence between the two errors, we found both errors of the estimation to be 15 cm.

To obtain the training data, the "true" location was not determined manually; instead, MCL was applied for position estimation (with a known starting position and very large sample sets). Empirical results in [20] suggest that MCL is sufficiently accurate for tracking a robot with only a few centimeters error. The robot's positions, while moving at speeds like 30 cm/sec through our environment, were synchronized and then further analyzed geometrically to determine whether (and where) the robot is in the visual fields of environmental cameras. As a result, data collection is extremely easy as it does not require any manual labeling; however, the error in MCL leads to a slightly less confined joint detection model than one would obtain with manually labeled data (assuming that the accuracy of manual position estimation exceeds that of MCL).

3.2. Negative Detections

Most of the techniques of state estimation use a sensor model that update the state belief when the sensor reports a measurement. However it is possible to get useful information of the state from the absence of environmental sensor measurements. There are two main reasons for environmental camera not to measure the robot marker. The first one is that the robot marker is not in the field of view of the environmental camera and the second one is that the environmental camera is unable to detect the robot mark, due to occlusions.

This situation of no detecting a robot mark can be modeled by considering the environmental camera field of view and by using an obstacle detection to identify occlusions as shown:

$$Bel(L^{(r)} = l) \leftarrow Bel(L^{(r)} = l) T(r_-^{(m)} | L^{(r)} = l, \mathbf{v}^{(m)}, \text{obs}^{(m)}) \quad (5)$$

where

$$T(r_-^{(m)} | L^{(r)} = l, \mathbf{v}^{(m)}, \text{obs}^{(m)}) = \begin{cases} 0 & l \in \mathbf{v}^{(m)} \text{ and } l \notin \text{obs}^{(m)} \\ 1 & l \notin \mathbf{v}^{(m)} \text{ or } l \in \text{obs}^{(m)} \end{cases},$$

$r_-^{(m)}$ represents the negative information of m -th environmental sensor, $\mathbf{v}^{(m)}$ describes the visibility area of the sensor and $\text{obs}^{(m)}$ represents the occlusion area.

The negative information has been applied to target tracking using the event of not detecting a target as evidence to update the probability density function [21]. In that work negative information means that the target is not located in the visual area of the sensor and since the target is known to exist it is certainly outside the area.

In robot localization domain, the work of Hoffmann *et al.* [22] on negative information in ML considers negative information the absence of landmark sensor measurements. Occlusions are identified using a visual sensor that scans colors of the ground to determine if there is free area or obstacle. The environment is soccer field in green with white lines. So, if a different color is identified, it means that an obstacle could be occluding the visibility of a landmark.

In the cooperative distributed sensor localization problem for mobile robot, negative information can also mean the absence of detections (in the case that an envi-

ronmental sensor does not detect the robot), which configures a lack of group information. In this case, the negative detection measurement can provide the useful information that the robot is not located in the visibility area of an environmental sensor. In some cases, it can be essential information as it could improve the pose belief of the robot in short time.

Our contribution in this paper is the proposal of a negative detection model and its incorporation into MCL approach based on distributed sensors. Consider an environmental camera, within a known environmental and its field of view as shown in **Figure 4(a)**. If the environmental camera does not detect the robot, negative information is reported, which states that the robot is not in the visibility area of the camera, as depicted in **Figure 4(a)**.

The information gathered from **Figure 4(a)** is true if we consider that there are no occlusions. In order to account for occlusions it is necessary to sense the environment to identify free areas or occupied areas. For environmental cameras, we apply background subtraction approach described in [23] to detect the occupied areas. If it is identified as an occupied area it means that the robot could be occluded by an obstacle. In this case, it is possible to use geometric inference to determine which part of the visual area can be used as negative detection information, as shown in **Figure 4(b)**.

3.3. Cooperative Distributed Sensor Localization

According to above positive and negative information, the cooperative distributed sensor localization algorithm for robot is summarized in **Table 1**.

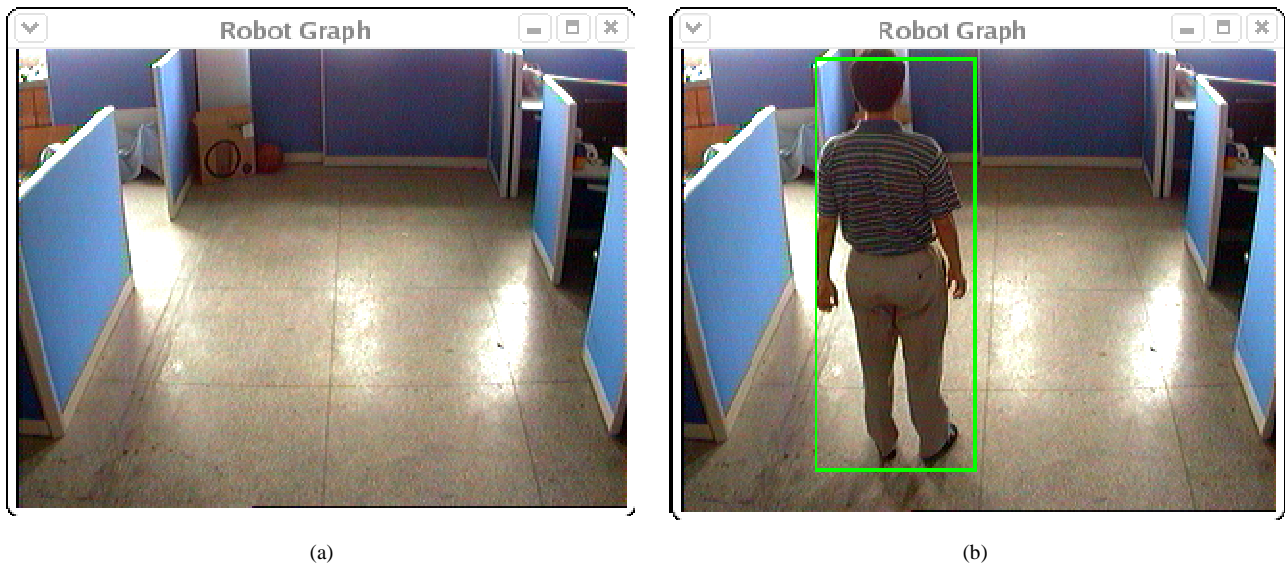


Figure 4. (a) Negative information; (b) Occlusion in the field of view.

Table 1. MCL algorithm to cooperate distributed sensors.

for each location l do	<i>/*initialize the belief*/</i>
$Bel^{(r)}(l) \leftarrow P(L_0^{(r)} = l)$	
end for	
forever do	
if the robot receives new laser inputs o do	<i>/*apply the laser perception model*/</i>
for each location l do	
$Bel^{(r)}(l) \leftarrow \alpha P(o l) Bel^{(r)}(l)$	
end for	
end if	
if the robot receives new odometry readings a do	<i>/*apply the motion model*/</i>
for each location l do	
$Bel^{(r)}(l) \leftarrow \sum_{l'} P(l a, l') Bel^{(r)}(l')$	
end for	
end if	
if the robot receives positive information from the m -th environmental sensor do	<i>/*apply the positive detection model*/</i>
for each location l do	
$Bel^{(r)}(l) \leftarrow Bel^{(r)}(l) \sum_{l'} P(L^{(r)} = l L^{(m)} = l', r_+^{(m)}) Bel^{(m)}(l')$	
end for	
end if	
if the robot receives negative information from the m -th environmental sensor do	<i>/*apply the negative detection model*/</i>
for each location l do	
$Bel(L^{(r)} = l) \leftarrow Bel(L^{(r)} = l) T(r_-^{(m)} L^{(r)} = l, \mathbf{v}^{(m)}, \text{obs}^{(m)})$	
end for	
end if	
end forever	

4. Experimental Results

In this section we present experiments conducted with real robot. The mobile robot used is Pioneer3 DX, which is equipped with a laser sensor. In whole experiments, the number N of samples is fixed to 400. **Figure 5(a)** shows our experimental setup along with a part of the occupancy grid map used for position estimation, and that two cameras are fixed on the wall applied to detect and localize the robot. **Figure 5(a)** also shows the visual fields of the two environmental cameras (the purple rectangle regions) and the path from A to C taken by Pioneer 3 DX with laser sensor, which was in the process of global localization. **Figure 5(b)** represents the uncertain belief of the robot on point A from scratch.

In order to evaluate the benefits of collaborative distributed sensor localization for robot, three different types of experiment are performed using the above deployment. The first one is that the robot performs global localization by using the positive information of environmental cameras, and the field of each camera is not occupied. The second one is to use positive and negative information of environmental cameras whose fields of view are not occupied for robot localization. Compared with the second one, the only difference of the last one is

that visual area of camera one is partly occupied.

4.1. No Occlusions and Only Using Positive Information

Before robot passes point B (shown in **Figure 6(a)**), the robot is still highly uncertain about its exact location only depending on its onboard laser sensor. The key event, illustrating the utility of cooperation in localization, is a detection event. More specifically, the environmental camera 1 detects the robot as it moves through its visual field (see **Figure 7**). Using the joint detection model described in Section 3, the robot integrates the positive information into its current belief. The effect of this integration on robot's belief is shown in **Figure 6(b)**. As this figure illustrates, this single incident almost completely resolves the uncertainty in robot's belief and shortens the time of robot global localization effectively.

4.2. No Occlusions and Using All Information

It can be seen from **Figure 8(a)** that the particles existing in the visibility area of two cameras are disappeared due to using the negative information. After five seconds, the effect of this integration on robot's belief is shown in

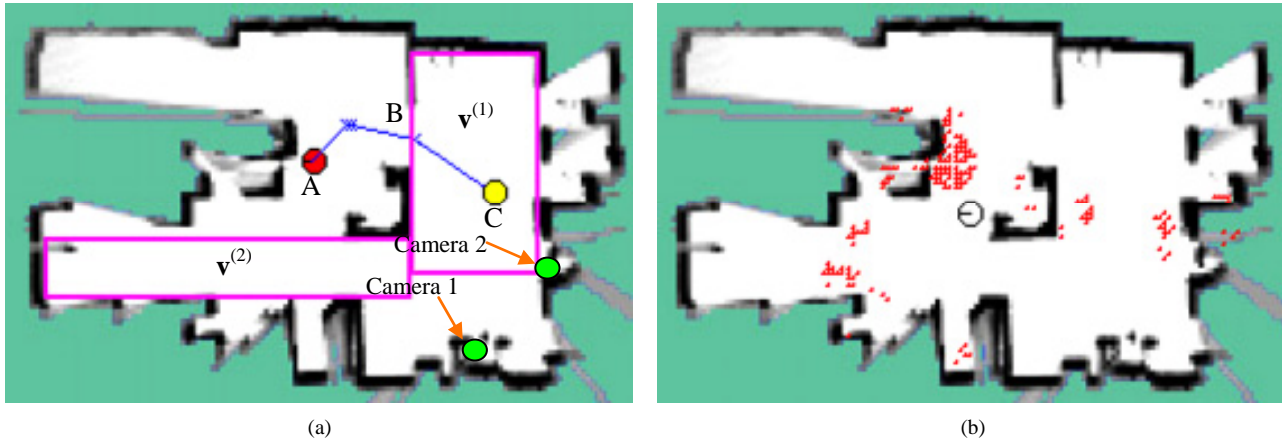


Figure 5. (a) Experimental setup; (b) The sample cloud represents the robot's belief on point A from scratch.



Figure 6. (a) Sample set before passing point B; (b) Achieved localization by integrating the positive information of camera 1.

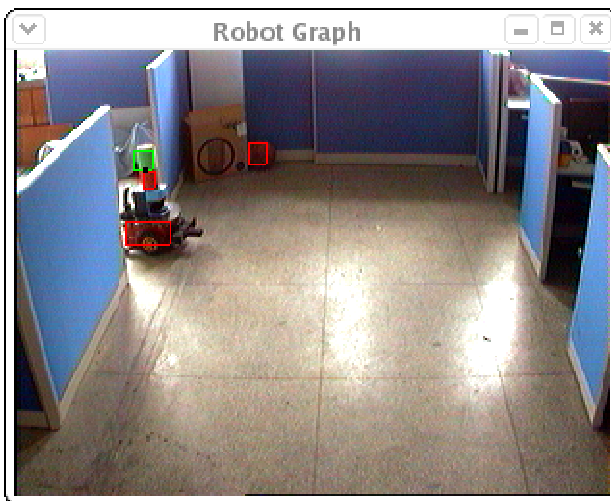


Figure 7. Detection event of camera 1 on robot passing point B.

Figure 8(b). Compared with experiment one, Localization results obtained with negative detection information into robot global localization are more accurate and provide the ability to localize robot more quickly.

4.3. Occlusions and Using All Information

In this experiment, we take into account the camera one being occupied by a people. The camera one applies background subtraction approach described in [23] to detect the occupied areas. The detection result is shown in **Figure 4(b)**. Due to the occlusion, the particles existing in the occupied areas are still reserved (see **Figure 9(a)**). After ten seconds, the effect of robot's belief is described in **Figure 9(b)**. From the experiment, it can be seen that though the camera one is partly occupied, the accuracy of the localization is still greatly improved using the negative detection information compared with experiment one.

4.4. Localization Error Analysis

In the case of no occlusions, we conducted ten times for the first two experiments and compared the performance to conventional MCL for robot which ignores environmental cameras' detections. To measure the performance to localization we determined the true locations of the

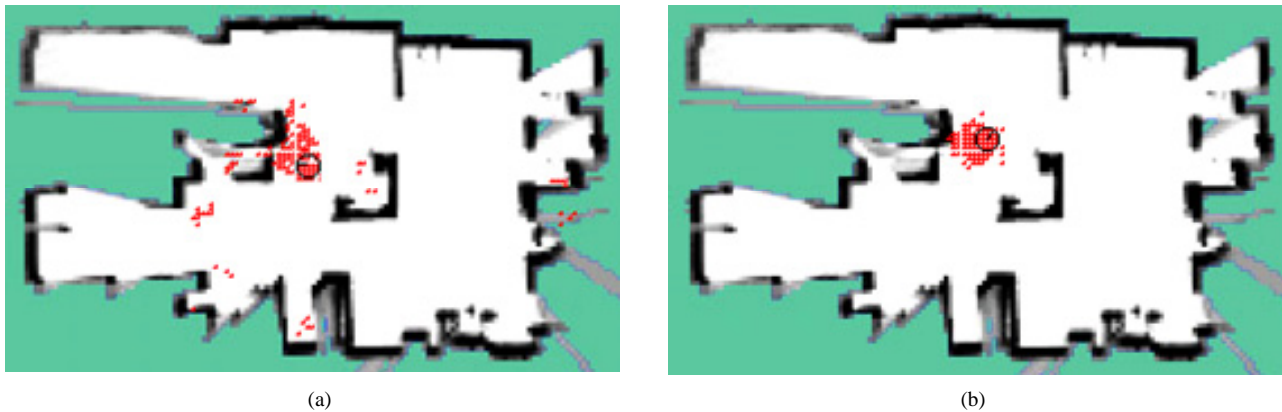


Figure 8. (a) The particles represent the robot's belief by integrating the negative information of two cameras; (b) Archived localization after five seconds.

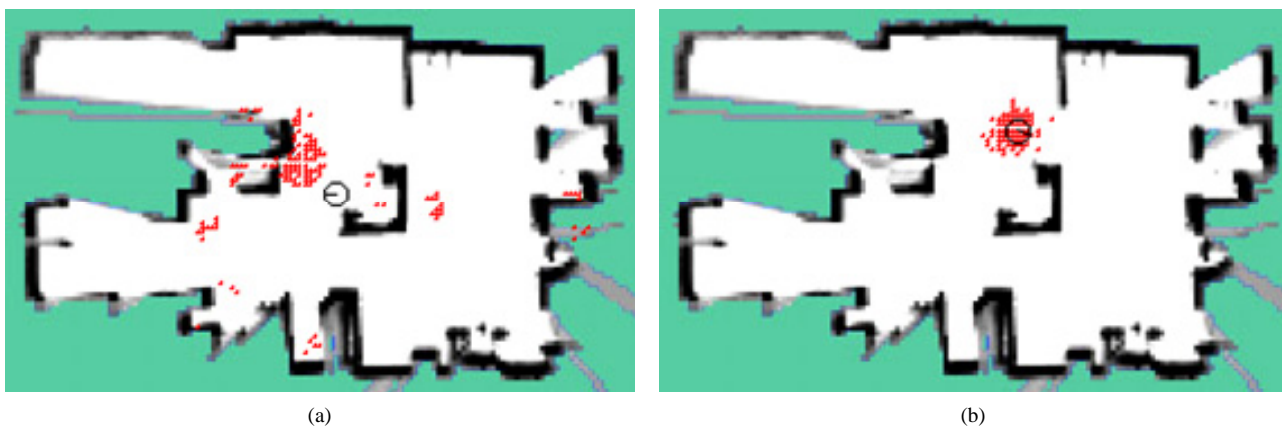


Figure 9. (a) Particles set after integrating negative information of two cameras among which the camera 1 is occupied by a people; (b) Archived localization after 10 seconds.

robot by performing position tracking and measuring the position of each second. For each second, we then computed the estimation error at the reference positions. The estimation error is measured by the average distance of all samples from the reference position. The results are summarized in **Figure 10**. The graph plots the estimation error (y-axis) as a function of time (x-axis), averaged over the ten experiments, along with their 95% confidence intervals (bars). Firstly, as can be seen in the figure, the quality of position estimation increases faster when using environmental camera detection (positive information) than one without environmental cameras. Please note that the detection event typically took place 14–16 seconds after the start of each experiment. Secondly, as can be also seen in the figure, the quality of position estimation increases much faster when using all information of environmental cameras. Obviously, this experiment is specifically well-suited to demonstrate the advantage of positive and negative information of environmental cameras in robot global localization. Of course, the performance of our approach in more complex situations, especially highly symmetrical and dy-

namic environments, is more attractable to solve robot's global localization.

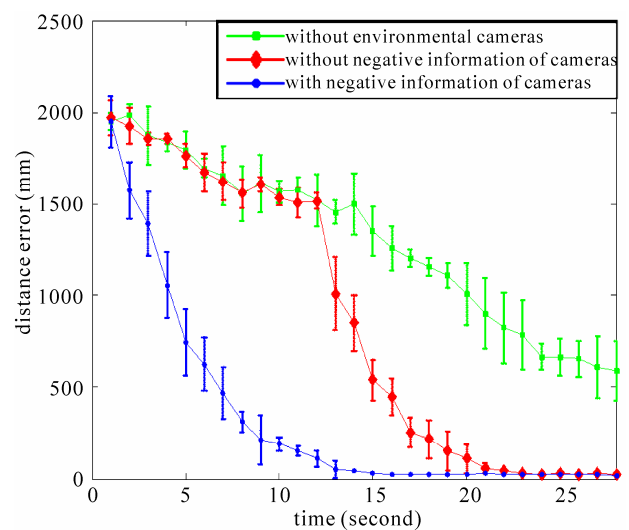


Figure 10. Comparison of localization error using three localization algorithms.

5. Conclusions

In this paper we presented an approach to collaborate distributed sensors for mobile robot localization that uses a sample-based representation of the state space of a robot, resulting in an extremely efficient and robust technique for global position estimation. Here we use environmental cameras whose parameters is unknown in advance to determine robot's position. In order to apply environmental cameras to localize the robot, all parameters of each environmental camera are calibrated independently by robot. During calibration, the robot localization is known and can navigate by its onboard laser sensor. Once calibrated, the positive and negative detections of the environmental cameras can be applied to localize robot. Experimental results demonstrate that, to combine all information of environmental cameras, the robot's belief can reduce its uncertainty significantly.

6. Acknowledgements

This work was supported by the Program of Nanjing University of Posts and Telecommunications (Grant No. NY209020).

7. References

- [1] N. Roy, W. Burgard, D. Fox and S. Thrun, "Coastal Navigation: Robot Navigation under Uncertainty in Dynamic Environments," *Proceedings of IEEE International Conference on Robotics and Automation*, Detroit, 10-15 May 1999, pp. 35-40.
- [2] S. Thrun, D. Fox, W. Burgard and F. Dellaert, "Robust Monte Carlo Localization for Mobile Robots," *Artificial Intelligence*, Vol. 128, 2001, pp. 99-141.
- [3] I. J. Cox and G. T. Wilfong, "Autonomous Robot Vehicles," Springer Verlag, Berlin, 1990.
- [4] T. Sogo, K. Kimoto, H. Ishiguro and T. Ishida, "Mobile Robot Navigation by a Distributed Vision System," *Journal of Japan Robotics Society*, Vol. 17, 1999, pp. 1009-1016.
- [5] K. Morioka, J. H. Lee and H. Hashimoto, "Human Centered Robotics in Intelligent Space," *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, Washington, D.C., 11-15 May 2002, pp. 2010-2015.
- [6] K. Matsumoto, H. Y. Chen, J. Ota and T. Arai, "Automatic Parameter Identification for Cooperative Modular Robots," *Proceedings of the 2002 IEEE International Symposium on Assembly and Task Planning*, 2002, pp. 282-287.
- [7] L. P. Kaelbling, A. R. Cassandra and J. A. Kurien, "Acting under Uncertainty: Discrete Bayesian Models for Mobile Robot Navigation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, 4-8 November 1996, pp. 963-972.
- [8] W. Burgard, D. Fox, D. Hennig and T. Schmidt, "Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids," *Proceedings of the National Conference on Artificial Intelligence*, Portland, 4-8 August 1996, pp. 896-901.
- [9] D. Fox, W. Burgard and S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," *Journal of Artificial Intelligence Research*, Vol. 11, 1999, pp. 391-427.
- [10] D. Fox, W. Burgard, F. Dellaert and S. Thrun, "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," *Proceedings of the National Conference on Artificial Intelligence*, Orlando, 1999, pp. 343-349.
- [11] D. B. Rubin, "Using the SIR Algorithm to Simulate Posterior distributions," In: M. H. Bernardo, K. M. De Groot, D. V. Lindley and A. F. M. Smith, Ed., *Bayesian Statistics 3*, Oxford University Press, Oxford, 1988.
- [12] A. F. M. Smith and A. E. Gelfand, "Bayesian Statistics Without Tears: A Sampling-Resampling Perspective," *American Statistician*, Vol. 46, No. 2, 1992, pp. 84-88.
- [13] N. J. Gordon, D. J. Salmond and A. F. M. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEEE Proceedings*, Vol. 140, No. 2, 1993, pp. 107-113.
- [14] M. Isard and A. Blake, "Condensation - Conditional density Propagation for Visual Tracking," *International Journal of Computer Vision*, Vol. 29, No.1, 1998, pp. 5-28.
- [15] K. Kanazawa, D. Koller and S. J. Russell, "Stochastic Simulation Algorithms for Dynamic Probabilistic Networks," *Proceedings of the 11th Annual Conference on Uncertainty in AI*, Montreal, 18-20 August 1995.
- [16] A. Doucet, "On Sequential Simulation-Based Methods for Bayesian Filtering," Technical Report CUED/FINFENG/TR.310, Department of Engineering, University of Cambridge, Cambridge, 1998.
- [17] H. Chen, K. Matsumoto, J. Ota and T. Arai, "Self-Calibration of Environmental Camera for Mobile Robot Navigation," *Robotics and Autonomous Systems*, Vol. 55, No. 3, 2007, pp. 177-190.
- [18] R. Y. Tsai, "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, 22-26 June 1986, pp. 364-374.
- [19] A. Howard and M. J. Mataric, "Localization for Mobile Robot Teams Using Maximum Likelihood Estimation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, 30 September-4 October 2002, pp. 434-459.
- [20] D. Koller and R. Fratkina, "Using Learning for Approximation in Stochastic Processes," *Proceedings of the International Conference on Machine Learning*, Madison, 24-27 July 1998, pp. 287-295.
- [21] W. Koch, "On Negative Information in Tracking and Sensor Data Fusion: Discussion of Selected Examples,"

- 7th International Conference on Information Fusion (FUSION 2004)*, Stockholm, 28 June-1 July 2004, pp. 91-98.
- [22] J. Hoffmann, M. Spranger, D. Gohring and M. Jungel, "Making Use of What you don't See: Negative Information in Markov Localization," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, 2-6 August 2005, pp. 854-859.
- [23] A. Mittal and N. Paragios, "Motion-Based Background Subtraction Using Adaptive Kernel Density Estimation," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, Washington, D.C., 27 June-2 July 2004, pp. 302-309.