

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Improved Monte Carlo Localization Algorithm in a Hybrid Robot and Camera Network

¹Zhiwei Liang, ²Xudong Ma, ²Fang Fang and ¹Songhao Zhu

¹College of Automation, Nanjing University of Posts and Telecommunications, Nanjing, 210046, China

²Key Lab of MCCSE, Ministry of Education, Southeast University, Nanjing, 210096, China

Abstract: In order to overcome the difficulty of a mobile robot to perform localization only with its onboard sensors, this study presents a probabilistic algorithm Monte Carlo Localization (MCL) to solve the problem of mobile robot localization in a hybrid robot and camera network in real time. On one hand, the robot does perform localization depending on its laser sensor using Monte Carlo method. On the other hand, environment cameras can detect the robot in their field of view during robot localization. According to a built environment camera model, MCL method extended to update robot's belief whichever information (positive or negative) attained from environmental camera sensors. Meanwhile, 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.

Key words: Environmental camera, monte carlo, negative information, positive information, global localization, kidnapped robot

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 (Roy *et al.*, 1999; Yuan *et al.*, 2009; Hassanzadeh and Mehdi, 2008) 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 (Liang *et al.*, 2008a). In recent years, a great lot of publications on localization have revealed the importance of the problem. Moreover, it has been referred to as the most fundamental problem to providing a mobile robot with autonomous capabilities.

Many existing work addresses localization problem only using sensors onboard mobile robots. However, during the process of 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 (Li and Zhang, 2007). In this case, various sensors are embedded into the environment (environmental sensors) and communication between the robot and environmental sensors is utilized. Both Lee *et al.* (2004) and Liang *et al.* (2008b) 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.* (2002) 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. Matsumoto *et al.* (2002) also proposed a concept called a distributed modular robot system. 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 structure 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 that a sensor detects the robot) to localize the 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 original work of Hoffmann *et al.* (2005) considered negative information in Markov localization for the Robocup Sony Aibo ERS-7 4-Legged robot. Experimental results show that integrating negative evidence can improve overall robot localization performance.

Inspired by the work of Hoffmann, the aim of this paper is to show how positive and negative information of sensors can be applied for robot localization in distributed sensor networks. Therefore, an efficient probabilistic approach based on Markov localization (Kaelbling *et al.*, 1996; Burgard *et al.*, 1996; Fox *et al.*, 1999) is proposed. In contrast to previous research, which relied on grid-based or coarse-grained topological representations of the robot state space, our approach adopts a sampling-based representation (Thrun *et al.*, 2001). 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 the robot. Once getting the cameras' parameters, the positive and negative detection models can be attained. Experimental results, carried out with three environmental cameras fixed in an indoor environment, illustrate the appropriateness of the approach in solving the robot global localization problem.

MONTE CARLO LOCALIZATION

Here, we will introduce our sampling-based localization approach only depending on the robot itself. It is based on Markov localization, which provides a general framework for estimating the position of a mobile robot. Markov localization maintains a belief $Bel(L^{(t)})$ over the complete three-dimensional state space of the robot. Here, $L^{(t)}$ denotes a random variable and $Bel(L^{(t)} = l)$ 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 (Thrun *et al.*, 2005; Smith and Gelfand, 1992). 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 (Gordon *et al.*, 1993), the Monte-Carlo filter (Thrun *et al.*, 2001), the Condensation algorithm (Isard and Blake, 1998), or the survival of the fittest algorithm (Kanazawa *et al.*, 1995). All these methods are generically known as particle filters and a discussion of their properties can be found in Doucet (1998).

More specifically, MCL represents the posterior beliefs $Bel(L)$ over the robot's state space by a set of N weighted random samples denoted $S = \{s_i | i = 1..N\}$. 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:

- 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
- 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:

$$p \leftarrow \alpha P(o|l)p \tag{1}$$

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 $P(o|o_i)$, which is the probability of observing o conditioned on the expected measurement o_i at location l . The expected measurement, a distance in this case, is easily computed from the map of the environment using ray tracing. The function is mixture of a Gaussian (centered on the correct distance o_i), a Geometric distribution (modeling overly short readings) and a Dirac distribution (modeling max-range readings) (Thrun *et al.*, 2005). 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 (Fox *et al.*, 1999).

COOPERATIVE DISTRIBUTED SENSOR LOCALIZATION

Here, 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.

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

$$Bel^{(t)}(l) \leftarrow \beta Bel^{(t)}(l) P(r_+^{(m)} | L^{(t)} = l) \tag{2}$$

where, β is normalization constant. The crucial component is the probabilistic detection mode:

$$P(r_+^{(m)} | L^{(t)} = l)$$

which describes the conditional probability that robot is at location l , given that sensor m perceives robot with positive measurement $r_+^{(m)}$. The detection model:

$$P(r_+^{(m)} | L^{(t)} = l)$$

of each environmental camera is constructed directly according to its parameters. Thus, before integrating positive information of each environmental camera into the robot belief, parameters of each camera need to be calibrated by the robot.

Camera self-calibration: In our method, all parameters of every environmental camera are unknown in advance and their Fields of View (FOV) are not overlaid each other. So, in order to apply them to localize the robot, every 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 (Chen *et al.*, 2007) into the system and take the mobile robot as a calibration instrument. Because the FOVs of all cameras are not overlaid each other, each camera calibration process is independent.

During the calibration, the robot location is known. When the robot moves depending on its laser and odometry in FOV of any environmental camera, the camera does detect the robot and gathers the relative data between the robot global location and detected image pixels. 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 0.1m. Once the number of relative data sums up to a threshold which is set as 200 in this paper, the camera calibration program can be conducted. Because the mobile robot always moves in a plane, the coplanar camera calibration method of Tsai and Roger (1986) is adopted here.

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 FOV of every camera moves as a zigzag, which is shown in Fig. 1.

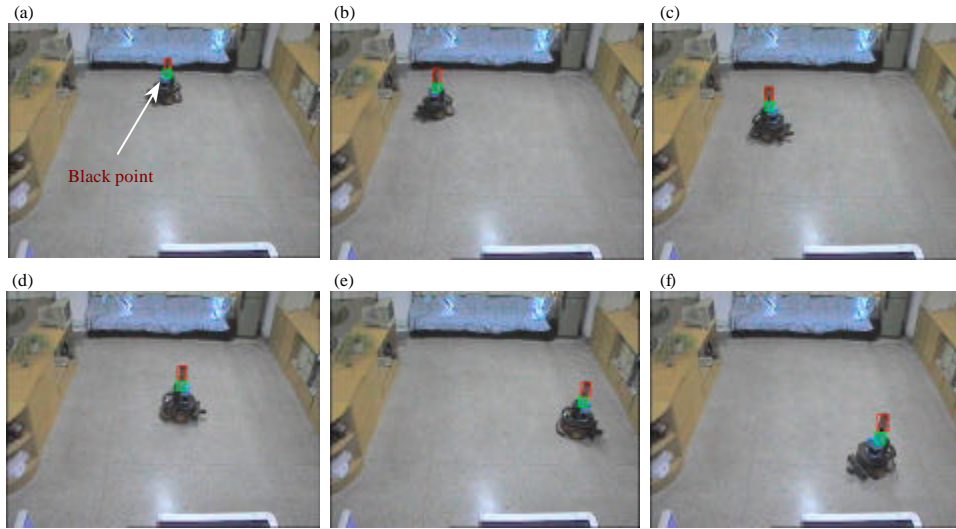


Fig. 1: Image sequences of successful detecting the robot which zigzagged through the visual field of a camera to calibrate the camera

Detection: To determine the location of the robot, our approach combines visual information obtained from environmental cameras. Camera images are used to detect the mobile robot and determine the position of the detected robot. The two rows in Fig. 1 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 shown with a flexible orientation in this Fig. 1, the marker can be detected regardless of the robot's orientation.

To find the robot in a camera image, our approach first 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 Fig. 1, shows the center of the marker as identified by this distributed environmental camera. Table 1 shows the ratios of false-positives and false-negatives estimated from a training set 120 images, in half of which the robot is within a camera FOV. As can be seen, our current visual routines have a 2.5% chance of not detecting a robot in the camera FOV and a 6.7% chance of detecting a robot which is not in the 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.

Detection model: Here, we have to devise a detection model of the type:

$$P(r_+^{(m)} | L^{(r)} = I)$$

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 estimated locations 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 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 the robot is in its FOV, 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 detection model:

$$P(r_+^{(m)} | L^{(r)} = I) = (2\pi)^{-\frac{1}{2}} \left| \Sigma_{I^{(m)}} \right|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} (AI - r_+^{(m)})' \Sigma_{I^{(m)}}^{-1} (AI - r_+^{(m)}) \right\} \quad (3)$$

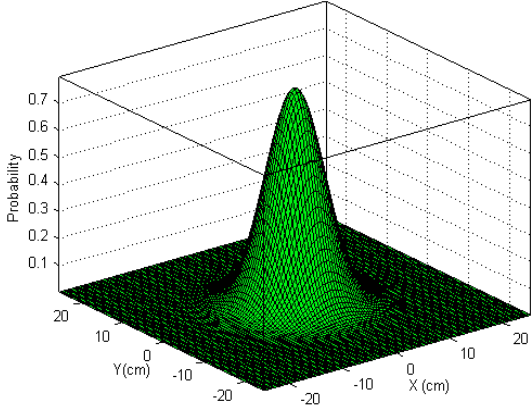


Fig. 2: Camera detection Gaussian model

where,

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \Sigma_{4+} = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$$

σ_x^2 and σ_y^2 represent mean square error in x direction and y direction respectively. Let, r_+ as coordinate origin and the Gaussian model showed in Fig. 2 models the error in the estimation of robot's location. Here, the x-axis represents the error of x direction in the world coordinates and the y-axis the y direction error. The parameters of this Gaussian model has been obtained through maximum likelihood estimation (Howard and Mataric, 2002) 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 about 10 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 by Koller and Fratkin (1998) 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⁻¹ through our environment, were synchronized and then further analyzed geometrically to determine whether (and where) the robot is in the FOVs 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 detection model than one would obtain with manually labeled data (assuming that the accuracy of manual position estimation exceeds that of MCL).

Table 1: Rates of false-positives and false-negatives for our detection routine

	Robot detected (%)	No robot detected (%)
Robot in camera FOV	97.5	2.5
No robot in camera FOV	6.7	93.3

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 three main reasons for environmental camera not to measure the robot marker. The first one is that the robot marker is not in the FOV of the environmental camera, the second one is that the environmental camera fails to detect the robot mark even if it falls within the camera FOV and the last 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 FOV and by using an obstacle detection to identify occlusions as shown:

$$Bel(L^{(t)} = l) \leftarrow \gamma Bel(L^{(t)} = l) P(r_+^{(m)} | L^{(t)} = l, v^{(m)}, obs^{(m)}) \quad (4)$$

where, γ is normalization constant, the negative detection model is defined as:

$$P(r_+^{(m)} | L^{(t)} = l, v^{(m)}, obs^{(m)}) = \begin{cases} 1 - (2\pi)^{-\frac{1}{2}} |\Sigma_{L^{(m)}}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(Al - r_+^{(m)})^T \Sigma_{L^{(m)}}^{-1} (Al - r_+^{(m)})\right\} - P(e_l | l) & l \in v^{(m)} \text{ and } l \notin obs^{(m)} \\ 1 & l \in v^{(m)} \text{ or } l \in obs^{(m)} \\ 0 & l \notin v^{(m)} \end{cases} \quad (5)$$

where, let:

$$\Sigma_{L^{(m)}}^{-1} = \Sigma_{L^{(m)}}^{-1},$$

$r_+^{(m)}$ represents the negative information of m-th environmental sensor, $v^{(m)}$ describes the visibility area of the sensor and $obs^{(m)}$ represents the occlusion area. $P(e_l | l)$ represents the probability that the environmental camera fails to detect the robot mark which falls within the camera FOV. According to Table 1, $P(e_l | l)$ is always equal to 0.025. Let, r_+ as coordinate origin, the Gaussian model showed in Fig. 3 models the error in the location estimation of no detecting the robot. Here, the x-axis represents the error of x direction in the world coordinates and the y-axis the y direction error.

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 (Koch, 2005). 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 the cooperative distributed sensor localization problem for mobile robot, negative information can also mean the absence of detections (in the case that an environmental 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 FOV as shown in Fig. 4. 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 at the bottom right image of Fig. 4.

The information gathered from Fig. 4 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. 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. For environmental cameras, we apply background subtraction approach (Liang *et al.*, 2008c) to detect the occupied areas. So, a rectangle area in camera image can be attained corresponding to the occupied area, as shown in the upper left image of Fig. 4. Then the real field $obs^{(m)}$ of the occupied area in the world coordinates is computed based on camera calibrated parameters. The intersection area of the $obs^{(m)}$ and the camera FOV $v^{(m)}$ is shown in the top right image of Fig. 4.

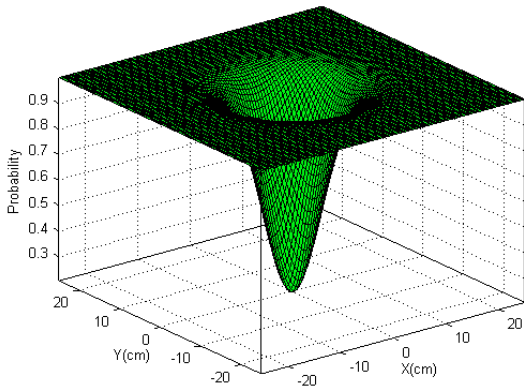


Fig. 3: Negative detection model

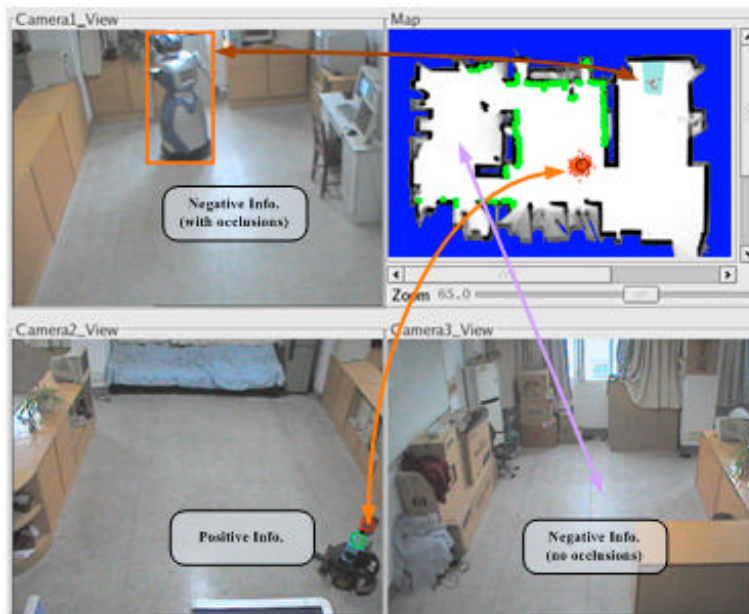


Fig. 4: Positive and negative detections

Cooperative distributed sensor localization: According to the above positive and negative information, the cooperative distributed sensor localization algorithm for robot is shown as follow:

MCL algorithm to cooperate distributed sensors

```

Step 1: Initialize the sample set  $S = \{s_i | i = 1 \dots N\}$  from  $P(L_n^{(0)} = l)$ 
 $s_i = \langle l_i, p_i \rangle$  where,  $l_i = \langle x_i, y_i, \theta_i \rangle$  denotes a robot position and  $p_i$ 
is a weighting factor.
Step 2: Apply the motion model
  if the robot receives new odometry readings do
    for  $i = 1$  to  $N$  do
       $l_i \leftarrow P(l_i | a, l_i)$ 
    end for
  end if
Step 3: Apply the perception model
  if the robot receives new laser sensor inputs  $O$  do
    for  $i = 1$  to  $N$  do
       $p_i \leftarrow \alpha P(o | l_i) p_i$ 
    end for
  end if
Step 4: Apply positive and negative detection model of camera
  for  $m = 1$  to  $M$  //  $M$  denotes the total number of all
  environmental sensors
    if the robot is detected by the  $m$ -th environmental sensor do
      for  $i = 1$  to  $N$  do
         $p_i \leftarrow \beta P(r_+^{(m)} | L^{(i)} = l) p_i$ 
      end for
    end if
    if the robot receives negative information from the  $m$ -th
    environmental sensor do
      for  $i = 1$  to  $N$  do
         $p_i \leftarrow \gamma P(r_-^{(m)} | L^{(i)} = l, v^{(m)}, obs^{(m)}) p_i$ 
      end for
    end if
  end for
Step 5: Sampling importance resampling (SIR) process (the details shown
  as [4])
  for  $i = 1$  to  $N$  do
    draw  $s_i$  with probability  $\propto p_i$ 
    add  $s_i$  to  $S$ 
  end for
  
```

RESULTS

The experimental system (Fig. 5) is composed of one ActivMedia Pioneer3 DX with one SICK LMS200 laser range sensor as the mobile robot and a camera network including three Panasonic CCD cameras. The position of each camera mounted on the ceiling is shown in Fig. 6. Moreover, each camera is linked to P4 2.0 Ghz+512 M RAM PC running Fedora Core 10 by a BT848 card. The camera nodes communicate with the robot based on wireless local network over a communication protocol called IPC, developed by Simmons (2008) at Carnegie Mellon University. Figure 7 shows the system software framework. It can be seen from Fig. 7 that each camera node manages the robot detection and camera calibration, while the robot runs the collaborative localization algorithm.

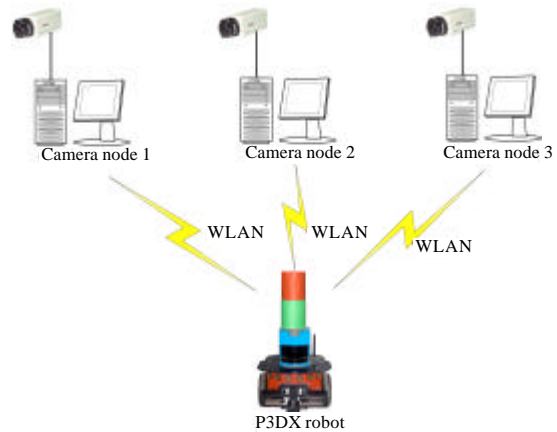


Fig. 5: The system hardware structure

In whole experiments, the number N of samples in cooperative distributed sensors localization algorithm is fixed to 400. Figure 7 is the system software interface including the occupancy grid map used for position estimation and the FOV of three cameras applied to detect and localize the robot. Figure 8 also shows the path from A to C taken by Pioneer3 DX with a laser sensor, which was in the process of global localization.

In order to evaluate the benefits of collaborative distributed sensor localization algorithm for the mobile 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 FOV of each camera is not occupied. The second one is to use positive and negative information of environmental cameras FOVs of which are not occupied for robot localization. Compared with the second one, the only difference of the last one is that FOV of each camera is partly occupied.

No occlusions and only using positive information:

Figure 9a shows the uncertain belief of the robot on point A from scratch. Before robot passes point B (shown in Fig. 9b), 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 2 detects the robot as it moves through its FOV at the 14th sec (Fig. 9c). Using the detection model described as earlier the robot integrates the positive information into its current belief. The effect of this integration on robot's belief is shown in Fig. 9c. As Figure 9c shows this single incident almost completely resolves the uncertainty in robot's belief and shortens the time of robot global localization effectively. When the

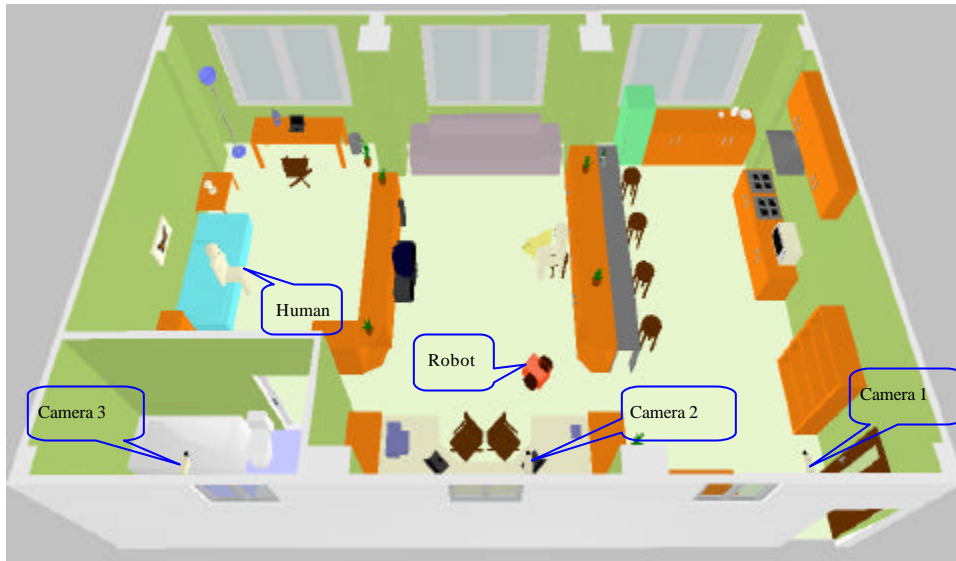


Fig. 6: 3D model of experimental environment

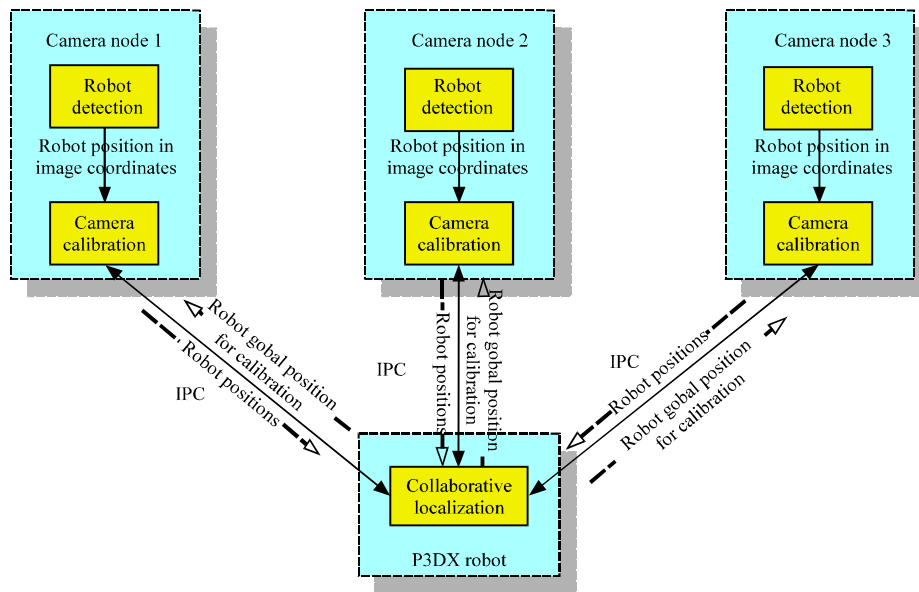


Fig. 7: The system software framework

robot arrived at point C, all samples have complete converge to the true pose of the robot, shown in Fig. 9d.

No occlusions and using all information: It can be seen from Fig. 10a that the particles existing in the visibility area of three cameras are disappeared due to using the negative information. After 9 sec, the effect of this integration on the robot's belief is shown in Fig.10b. Compared with the first experiment, localization results obtained with negative detection information into the

robot global localization are more accurate and provide the ability to localize the robot more quickly.

Occlusions and using all information: In this experiment, we take into account three cameras occupied by another three robots. The cameras applied background subtraction approach described by Liang *et al.* (2008c) to detect the obstacles. The detection result is shown in Fig. 11a. Due to the occlusion, the particles existing in the occupied areas are still reserved (Fig. 11b). After



Fig. 8: The software interface including the experimental map and FOV of three cameras

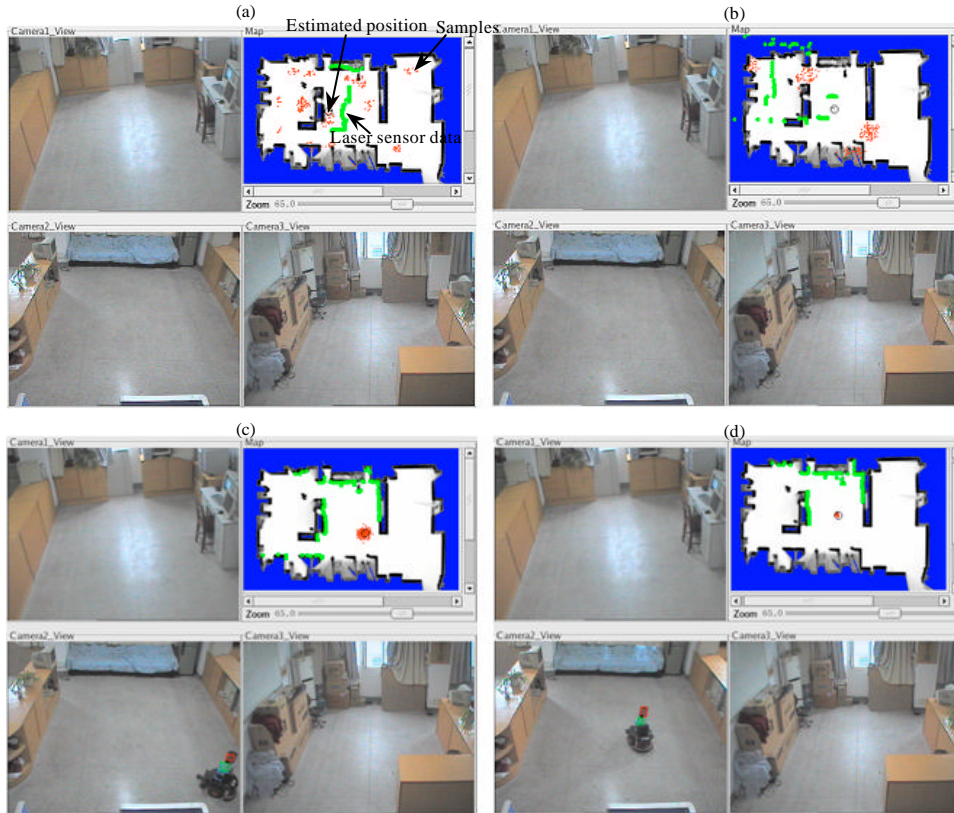


Fig. 9: The localization process using positive information of three cameras. (a) The sample cloud represents the robot's belief on point A from scratch, (b) Sample set before passing point B, (c) Achieved localization by integrating the positive information of camera 2 and (d) Sample set when the robot arrived at point C



Fig. 10: The localization process using all information of three cameras. (a) The particles represent the robot's belief by integrating the negative information of three cameras and (b) Archived localization after 8 sec

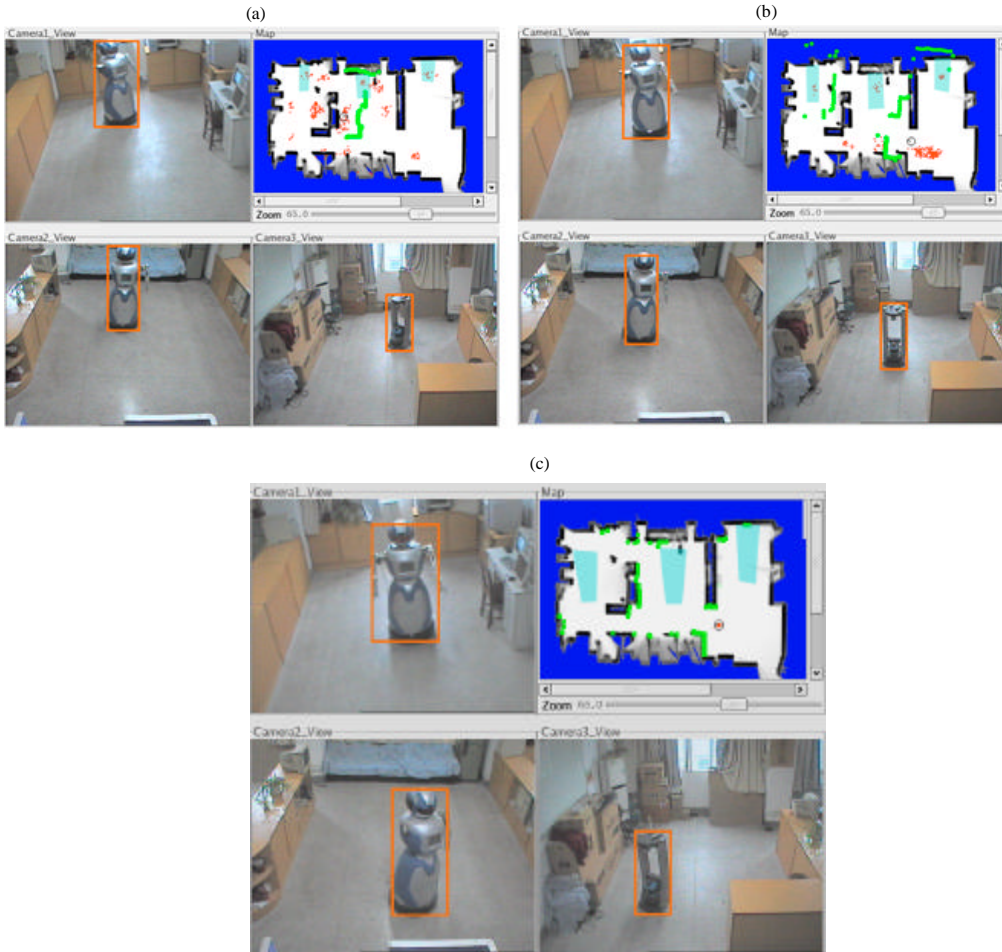


Fig. 11: The localization process using all information of three cameras the FOV of which were occupied by other three robots. (a) The sample cloud represents the robot's belief on point A from scratch, (b) Particles set after integrating positive and negative information of three cameras and (c) Archived localization after 10 sec

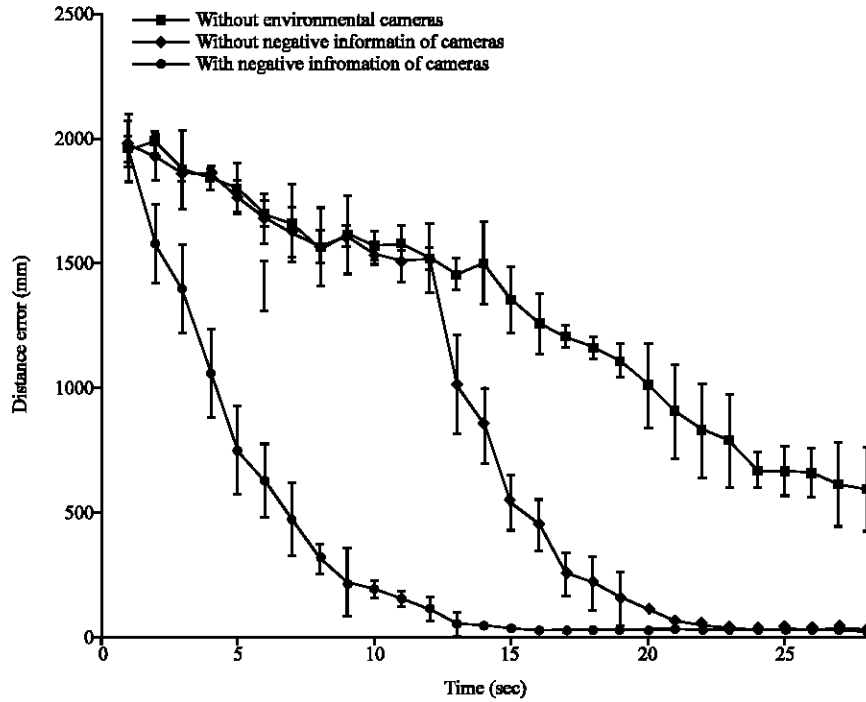


Fig. 12: Comparison of localization error using three localization algorithms

10 sec, the effect of robot's belief is described in Fig. 11c. From this experiment, it can be seen that though the cameras are partly occupied, the accuracy of the localization is still greatly improved using the negative detection information compared with the first experiment.

Localization error analysis: In the case of no occlusions, we conducted ten times for the first two experiments and another experiment that the robot performed global localization only using its laser sensor and compared the performance to conventional MCL for robot which ignores environmental cameras' detections. To measure the localization performance we determined the true locations of the 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 Fig. 12. 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 Fig. 12, 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 sec after the start of each

experiment and the robot resolves its global localization completely about at 18th sec. Secondly, as can be also seen in the Fig. 12, the quality of position estimation increases much faster (about 9 sec) 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 dynamic environments, is more attractable to solve robot's global localization.

CONCLUSIONS

In this study, we presented an approach to collaborate distributed sensors for mobile robot localization that uses a sample-based representation of the robot state space, 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.

ACKNOWLEDGMENTS

This study was supported by the National 863 Program of China (Grant No. 2007AA041703 and 2006AA040202), the Natural Science Foundation of China (Grant No. 60805032) and the Program of NJUPT (Grant No. NY209020 and NY209018).

REFERENCES

- Burgard, W., D. Fox, D. Henig and T. Schmidt, 1996. Estimating the absolute position of a mobile robot using position probability grids. Proceedings of the 13th National Conference on Artificial Intelligence, Aug. 04-08, Portland, Oregon, USA., pp: 896-901.
- Chen, H., K. Matsumoto, J. Ota and T. Arai, 2007. Self-calibration of environmental camera for mobile robot navigation. *Robotics Autonomous Syst.*, 55: 177-190.
- Doucet, A., 1998. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/FINFENG/TR.310. Department of Engineering, University of Cambridge.
- Fox, D., W. Burgard and S. Thrun, 1999. Markov localization for mobile robots in dynamic environments. *J. Artificial Int. Res.*, 11: 391-427.
- Gordon, N.J., D.J. Salmond and A.F.M. Smith, 1993. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F*, 140: 107-113.
- Hassanzadeh, I. and A.F. Mehdi, 2008. Design of augmented extended and unscented kalman filters for differential-drive mobile robots. *J. Applied Sci.*, 8: 2901-2906.
- Hoffmann, J., M. Spranger, D. Gohring and M. Jungel, 2005. Making use of what you don't see: Negative information in Markov localization. Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, (IRS'05), USA., pp: 2947-2952.
- Howard, A. and M.J. Mataric, 2002. Localization for mobile robot teams using maximum likelihood estimation. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sept. 30-Oct. 04, EPFL Switzerland, pp: 434-459.
- Isard, M. and A. Blake, 1998. Condensation-conditional density propagation for visual tracking. *Int. J. Comput. Vision*, 29: 5-28.
- Kaelbling, L.P., A.R. Cassandra and J.A. Kurien, 1996. Acting under uncertainty: Discrete bayesian models for mobile robot navigation. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nov. 4-8, Osaka, pp: 963-972.
- Kanazawa, K., D. Koller and S.J. Russell, 1995. Stochastic simulation algorithms for dynamic probabilistic networks. Proceeding of the 11th Annual Conference on Uncertainty in AI, (UAI'95), Montreal, Canada, pp: 346-351.
- Koch, W., 2005. On negative information in tracking and sensor data fusion: Discussion of selected examples. Proceeding of the 7th International Conference on Informatics in Control, Automation and Robotics, (ICAR'05), USA, pp: 88-93.
- Koller, D. and R. Fratkina, 1998. Using learning for approximation in stochastic processes. Proceedings of the 15th International Conference on Machine Learning, July 24-27, San Francisco, CA, USA., pp: 287-295.
- Lee, J.H., K. Morioka, N. Ando and H. Hashimoto, 2004. Cooperation of distributed intelligent sensors in intelligent environment. *IEEE/ASME Trans. Mechatronics*, 9: 535-543.
- Li, S. and D. Zhang, 2007. Distributed localization based on hessian local linear embedding algorithm in wireless sensor networks. *Inform. Technol. J.*, 6: 885-890.
- Liang, Z.W., X.D. Ma and X.Z. Dai, 2008a. Information-theoretic approaches based on sequential Monte Carlo to collaborative distributed sensors for mobile robot localization. *J. Intell. Robot Syst.*, 52: 157-174.
- Liang, Z.W., X.D. Ma and X.Z. Dai, 2008b. An extended Monte Carlo localization approach based on collaborative distributed perception. *Robot*, 30: 210-216.
- Liang, Z.W., X.D. Ma and X.Z. Dai, 2008c. Compliant navigation mechanisms utilizing probabilistic motion patterns of humans in a camera network. *Adv. Robotics*, 22: 929-948.
- Matsumoto, K., H.Y. Chen, J. Ota and T. Arai, 2002. Automatic parameter identification for cooperative modular robots. Proceedings of the IEEE International Symposium on Assembly and Task Planning, (SATP'02), USA., pp: 282-287.
- Morioka, K., J.H. Lee and H. Hashimoto, 2002. Human centered robotics in intelligent space. Proceedings of the IEEE International Conference on Robotics and Automation, (RA'02), USA., pp: 2010-2015.
- Roy, N., W. Burgard, D. Fox and S. Thrun, 1999. Coastal navigation: Robot navigation under uncertainty in dynamic environments. Proceedings of the IEEE International Conference on Robotics and Automation, (RA'99), Detroit, MI, pp: 35-40.

- Simmons, R., 2008. The Inter-process Communication (IPC) system. <http://www-2.cs.cmu.edu/afs/cs/project/TCA/www/ipc/ipc.html>.
- Smith, A.F.M and A.E. Gelfand, 1992. Bayesian statistics without tears: A sampling-resampling perspective. *Am. Statist.*, 46: 84-88.
- Thrun, S., D. Fox and W. Burgard, 2001. Robust monte carlo localization for mobile robots. *Artificial Intel.*, 128: 99-141.
- Thrun, S., W. Burgard and D. Fox, 2005. *Probabilistic Robotics*. The MIT Press, UK., ISBN-13: 978-0-262-20162-9.
- Tsai and Y. Roger, 1986. An efficient and accurate camera calibration technique for 3D machine vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR'86)*, Miami Beach, FL, pp: 364-374.
- Yuan, X., C.X. Zhao and Z.M. Tang, 2009. Lidar scan-matching for mobile robot localization. *Inform. Technol. J.*, 9: 27-33.