12

A Visual Based Extended Monte Carlo Localization for Autonomous Mobile Robots

Wen Shang[1] and Dong Sun[2]
[1]*Suzhou Research Institute of City University of Hong Kong*
[2]*Department of Manufacturing Engineering and Engineering Management of City University of Hong Kong*
P.R. China

1. Introduction

Over the past decades, there are tremendous researches on mobile robots aiming at increasing autonomy of mobile robot systems. As a basic problem in mobile robots, self-localization plays a key role in various autonomous tasks (Kortenkamp et al., 1998). Considerable researches have been done on self-localization of mobile robots (Borenstein et al., 1996; Chenavier & Crowley, 1992; Jensfelt & Kristensen, 2001; Tardos et al., 2002), with the goal of estimating the robot's pose (position and orientation) by proprioceptive sensors and exteroceptive techniques. Since proprioceptive sensors (e.g., dead-reckoning) are generally not sufficient to locate a mobile robot, exteroceptive techniques have to be used to estimate the robot's configuration more accurately. Some range sensors such as sonar sensors (Drumheller, 1987; Tardos et al., 2002; Wijk & Christensen, 2000) and laser range finders (Castellanos & Tardos, 1996), can be employed for the robot localization. However, the data obtained from sonar sensors is usually noisy due to specular reflections, and the laser scanners are generally expensive. As a result, other sensory systems with more reliable sensing feedback and cheaper price, such as visual sensors (Chenavier & Crowley, 1992; Dellaert et al., 1999; Gaspar et al., 2000), are more demanded for mobile robot localization.

Probabilistic localization algorithm (Chenavier & Crowley, 1992; Fox et al., 1999b; Nourbakhsh et al., 1995) is a useful systematic method in sensor-based localizations, providing a good framework by iteratively updating the posterior distribution of the pose space. As a state estimation problem, pose estimation with linear Gaussian distribution (unimodal) can be done by Kalman filters for pose tracking (Chenavier & Crowley, 1992; Leonard & Durrant-White, 1991), which exhibits good performance under the condition that the initial robot pose is known. Nonlinear non-Gaussian distribution (multimodal) problem can be solved by multi-hypothesis Kalman filters (Jensfelt & Kristensen, 2001) or Markov methods (Fox et al., 1999b; Nourbakhsh et al., 1995) for global localization. The multi-hypothesis Kalman filters use mixtures of Gaussians and suffer from drawbacks inherent with Kalman filters. Markov methods employ piecewise constant functions (histograms) over the space of all possible poses, so the computation burden and localization precision depend on the discretization of pose space.

Source: Vision Systems: Applications, ISBN 978-3-902613-01-1
Edited by: Goro Obinata and Ashish Dutta, pp. 608, I-Tech, Vienna, Austria, June 2007

By representing probability densities with sets of samples and using the sequential Monte Carlo importance sampling, Monte Carlo localization (MCL) (Dellaert et al., 1999; Fox et al., 1999a) represents non-linear and non-Gaussian models with great robustness and can focus the computational resources on regions with high likelihood. Hence MCL has attracted considerable attention and has been applied in many robot systems. MCL shares the similar idea to that of particle filters (Doucet, 1998) and condensation algorithms (Isard & Blake, 1998) in computer vision.

As a sample based method with stochastic nature, MCL can suffer from the observation deviation or over-convergence problem when the sample size is smaller or encountering some poorly modeled events (to be discussed in detail in Section 2.2) (Carpenter et al., 1999; Thrun et al., 2001). Many approaches have been proposed to improve the efficiency of MCL algorithm. A method of adaptive sample size varying in terms of the uncertainty of sample distribution, was presented in (Fox, 2003). However, the sample size of this method must meet a condition of an error bound of the distribution, which becomes a bottleneck for a real global localization. A resampling process through introduction of a uniform distribution of samples was further applied for the case of non-modeled movements (Fox et al., 1999a). Likewise, a sensor resetting localization algorithm (Lenser & Veloso, 2000) was also implemented using a resampling process from visual feedback, based on an assumption that the visual features with range and bearing are distinguishable. Such a method may be applicable to RoboCup, but not to a general office environment. Several other visual based Monte Carlo methods (Kraetzschmar & Enderle, 2002; Rofer & Jungel, 2003) were implemented under the condition that the environment features must be unique. A mixture MCL (Thrun et al., 2001) and condensation with planned sampling (Jensfelt et al., 2000) incorporated the resampling process to MCL for efficiency improvement, which require fast sampling rate from sensors every cycle.

In order to achieve higher localization precision and improve efficiency of MCL, a new approach to extended Monte Carlo localization (EMCL) algorithm is presented here. The basic idea is to introduce two validation mechanisms to check the abnormity (e.g., observation deviation and over-convergence phenomenon) of the distribution of weight values of sample sets and then employ a resampling strategy to reduce their influences. According to the verification, the strategy of employing different resampling processes is employed, in which samples extracted either from importance resampling or from observation model form the true posterior distribution. This strategy can effectively prevent from the premature convergence and be realized with smaller sample size. A visual-based extended MCL is further implemented. The common polyhedron visual features in office environments are recognized by Bayesian network that combines perceptual organization and color model. This recognition is robust with respect to individual low-level features and can be conveniently transferred to similar environments. Resampling from observation model is achieved by the triangulation method in the pose constraint region.

The remainder of this chapter is organized as follows. Section 2 introduces conventional MCL algorithm and discusses the existing problems when applied to the real situations. Section 3 proposes the extended MCL (EMCL) with brief implementation explanations showing the difference from conventional MCL, which is followed by the implementation details of a visual-based EMCL application example in Section 4. Section 5 presents experiments conducted on a mobile robot system to verify the proposed approach. Finally, conclusions of this work are given in Section 6.

2. Conventional Monte Carlo Localization

2.1 Conventional MCL

Monte Carlo localization (MCL) (Dellaert et al., 1999; Fox et al., 1999a) is a recursive Bayesian filter that estimates the posterior distribution of robot poses conditioned on observation data, in a similar manner to Kalman filters (Chenavier & Crowley, 1992) and Markov methods (Fox et al., 1999b; Nourbakhsh et al., 1995). The robot's pose is specified by a 2D Cartesian position x_k and y_k, and a heading angle θ_k, where *k* denotes the index of time sequences. It is assumed that the environment is Markov when using Bayesian filters, that is, the past and the future data are (conditionally) independent if one knows the current state. The iterative Markov characteristic of Bayesian filters provides a well probabilistic update framework for all kinds of probability-based localization algorithms.

MCL is implemented based on SIR (Sampling/Importance Resampling) algorithm (Carpenter et al., 1999; Doucet, 1998) with a set of weighted samples. For the robot pose $X_k = [x_k \quad y_k \quad \theta_k]^T$, define the sample set as follows:

$$S_k = \{s_k^{(i)} =< X_k^{(i)}, w_k^{(i)} > | \, i = 1, \cdots, N_k\}$$

where the sample $s_k^{(i)}$ consists of the robot pose $X_k^{(i)}$ and the weight $w_k^{(i)}$ that represents the likelihood of $X_k^{(i)}$, i is the index of weighted samples, and N_k denotes the number of samples (or sample size). It is assumed that $\sum_{i=1}^{N_k} w_k^{(i)} = 1$, since the weights are interpreted as probabilities.

During the localization process, MCL is initialized with a set of samples reflecting initial knowledge of the robot's pose. It is usually assumed that the distribution is uniform for global localization when the initial pose is unknown, and a narrow Gaussian distribution when the initial pose is known. Then samples are recursively updated with the following three steps executed (see Table 1).

Step 1: Sample update with robot motion (prediction step)

The probabilistic proposal distribution of robot pose in the motion update is

$$q_k = p(X_k \mid X_{k-1}, u_{k-1}) \times Bel(X_{k-1}) \tag{1}$$

where $p(X_k \mid X_{k-1}, u_{k-1})$ denotes probabilistic density of the motion that takes into account the robot properties such as drift, translational and rotational errors, $u_{k-1} = [\Delta x_{k-1} \quad \Delta y_{k-1} \quad \Delta\theta_{k-1}]^T$ denotes variation of the robot pose at time *k*-1, and $Bel(X_{k-1})$ denotes posterior distribution of the robot pose X_{k-1}. Then, extract a new sample set S_k' with $< \overline{X}_k^{(i)}, \overline{w}_k^{(i)} >$ from the proposal distribution q_k, by applying the above motion update to the posterior distribution, where $\overline{X}_k^{(i)}$ and $\overline{w}_k^{(i)}$ denote the extracted pose and weight after motion update, respectively.

Step 2: Belief update with observations (sensor update step)

Robot's belief about its pose is updated with observations, mostly from range sensors. Introduce a probabilistic observation model $p(Z_k \mid X_k^{(i)})$, where Z_k denotes measurements

from the sensor. Re-weight all samples of S_k' extracted from the prediction step, and we then have

Algorithm Conventional MCL

Prediction step:
for each $i = 1, \cdots, N_k$
Draw sample $\overline{X}_k^{(i)}$ from S_{k-1} according to (1)
$\overline{w}_k^{(i)} = 1 / N_k$
$< \overline{X}_k^{(i)}, \overline{w}_k^{(i)} > \rightarrow S_k'$
end for
Sensor update step:
for each $i = 1, \cdots, N_k$
$\hat{w}_k^{(i)} = \overline{w}_k^{(i)} \cdot p(Z_k \mid \overline{X}_k^{(i)})$
$\widetilde{w}_k^{(i)} = \dfrac{\hat{w}_k^{(i)}}{\sum_{j=1}^{N_k} \hat{w}_k^{(j)}}$
$< \overline{X}_k^{(i)}, \widetilde{w}_k^{(i)} > \rightarrow S_k''$
end for
Resampling step (importance resampling):
for each $\widetilde{s}_k^{(i)} =< \overline{X}_k^{(i)}, \widetilde{w}_k^{(i)} >$ in S_k''
$cw(\widetilde{s}_k^{(i)}) = \sum_{j=1}^{i} \widetilde{w}_k^{(j)}$ {Cumulative distribution}
end for
for each $i = 1, \cdots, N_k$
r=rand(0,1); {random number r}
j=1
while($j \le N_k$) do
if($cw(\widetilde{s}_k^{(j)}) > r$)
$X_k^{(i)} = \overline{X}_k^{(j)}$
$w_k^{(i)} = 1 / N_k$
$< X_k^{(i)}, w_k^{(i)} > \rightarrow S_k$, break
else j=j+1
end if
end while
end for

Table 1. Conventional MCL algorithm

$$\hat{w}_k^{(i)} = \overline{w}_k^{(i)} \cdot p(Z_k \mid \overline{X}_k^{(i)}) \tag{2}$$

where $\hat{w}_k^{(i)}$ denotes the non-normalized weight during the sensor update.
Normalize weights as follows to ensure that all beliefs sum up to 1:

$$\tilde{w}_k^{(i)} = \frac{\hat{w}_k^{(i)}}{\sum_{j=1}^{N_k} \hat{w}_k^{(j)}} \tag{3}$$

Then, the sample set after sensor update, denoted by S_k'' with $< \overline{X}_k^{(i)}, \tilde{w}_k^{(i)} >$, is obtained. The observation model $p(Z_k \mid X_k^{(i)})$ is also named as importance factor (Doucet, 1998), which reflects the mismatch between the probabilistic distribution q_k after the prediction step and the current observations from the sensor.

Step 3: Resampling step

The resampling step is to reduce the variance of the distribution of weight values of samples and focus computational resources on samples with high likelihood. A new sample set S_k is extracted with samples located nearby the robot true pose. This step is effective for localization by ignoring samples with lower weights and replicating those with higher weights. The step is to draw samples based on the importance factors, and is usually called importance resampling (Konolige, 2001). The implementation of such importance resampling is shown in Table 1.

2.2 Problems of Conventional MCL

When applied to the real situations, conventional MCL algorithm suffers from some shortcomings. The samples are actually extracted from a proposal distribution (here is the motion model). If the observation density deviates from the proposal distribution, the (non-normalized) weight values of most of the samples become small. This leads to poor or even erroneous localization result. Such phenomenon results from two possible reasons. One is that too small sample size is used, and the other is due to poorly modeled events such as kidnapped movement (Thrun et al., 2001). To solve the problem, either a large sample size is employed to represent the true posterior density to ensure stable and precise localization, or a new strategy is employed to address the poorly modeled events.

Another problem when using conventional MCL is that samples often converge too quickly to a single or a few high-likelihood poses (Luo & Hong, 2004), which is undesirable in the localization in symmetric environments, where multiple distinct hypotheses have to be tracked for periods of time. This over-convergence phenomenon is caused by the use of too small sample size, as well as smaller sensor noise level. The viewpoint that the smaller the sensor noise level is, the more likely over-convergence occurs, is a bit counter-intuitive, but it actually leads to poor performance. Due to negative influences of the smaller sample size and poorly modeled events, implementation of conventional MCL in real situations is not trivial.

Since sensing capabilities of most MCLs are achieved by sonar sensors or laser scanners, the third problem is how to effectively realize MCL with visual technology, which can more accurately reflect the true perceptual mode of the natural environments.

3. Extension of Monte Carlo Localization (EMCL)

In order to overcome limitations of conventional MCL when applied to real situations, a new approach to extended Monte Carlo localization (EMCL) methodology is proposed in this section.

In the proposed extended MCL algorithm, besides the prediction and sensor update steps that are the same as in the conventional MCL, two validation mechanisms in the resampling step are introduced for checking abnormity of the distribution of weight values of sample sets. According to the validation, different resampling processes are employed, where samples are extracted either from importance resampling or from observation model. Table 2 gives the procedures of the proposed extended MCL algorithm.

Algorithm Extended MCL

Prediction step:
Sensor update step:
Same as conventional MCL algorithm;
Resampling step: (different from conventional MCL)
Quantitatively describe the distribution of (normalized and non-normalized) weight values of sample set;
Two validation mechanisms:
if (over-convergence); **over-convergence validation**
sample size n_s resampling from observations
for each $i = 1, \cdots, N_k - n_s$
importance resampling $X_k^{(i)}$ from S_k''
$w_k^{(i)} = 1 / N_k$
$< X_k^{(i)}, w_k^{(i)} > \rightarrow S_k$
end for
for each $i = N_k - n_s + 1, \cdots, N_k$
sensor based resampling
$X_k^{(i)} \leftarrow p(X_k \mid Z_k)$
$w_k^{(i)} = 1 / N_k$
$< X_k^{(i)}, w_k^{(i)} > \rightarrow S_k$
end for
else if (sum of (non-normalized) weight $< W_{th}$); **uniformity validation**
resampling size $n_s = N_k$
sensor based resampling (same as the above)
else importance resampling
end if
end if

Table 2. Extended MCL algorithm

Two Validation Mechanisms

The two validation mechanisms are uniformity validation and over-convergence validation, respectively.

Uniformity validation utilizes the summation of all non-normalized weight values of sample set after sensor update to check the observation deviation phenomenon, in which the non-normalized weight values in the distribution are uniformly low, since the observation

density deviates from the proposal distribution due to some poorly modeled events.
Since the samples are uniformly distributed after the prediction step and re-weighted through the sensor update step, summation of non-normalized weight values of all samples W can be, according to (2), expressed as

$$W = \sum_{i=1}^{N_k} \hat{w}_k^{(i)} = \sum_{i=1}^{N_k} \overline{w}_k^{(i)} p(Z_k \mid \overline{X}_k^{(i)}) = \frac{1}{N_k} \sum_{i=1}^{N_k} p(Z_k \mid \overline{X}_k^{(i)}) \tag{4}$$

where, N_k denotes the sample size at time index *k*; $\overline{w}_k^{(i)}$ and $\hat{w}_k^{(i)}$ denote the weight values of sample $\overline{X}_k^{(i)}$ after motion update and after sensor update, respectively.

Define W_{th} as the given threshold corresponding to the summation of the weight values. If the summation W of all non-normalized weight values of samples is larger than the given threshold W_{th}, the observation can be considered to be consistent with the proposal distribution, and the importance resampling strategy is implemented. Otherwise, deviation of observations from the proposal distribution is serious, and the sensor-based resampling strategy is applied by considering the whole sample size at the moment as the new sample size. The given threshold should be appropriately selected based on the information of the observation model and the observed features.

Over-convergence validation is used to handle the over-convergence phenomenon, where samples converge quickly to a single or a few high-likelihood poses due to smaller sample size or lower sensor noise level. Over-convergence validation is employed based on the analysis of the distribution of normalized weight values of sample set, in which entropy and effective sample size are treated as measures for validation. When over-convergence phenomenon is affirmed, the strategy of both importance resampling and sensor-based resampling will be applied.

Entropy denotes the uncertainty of probabilistic events in the form of $H = -\sum p_i \log p_i$, where p_i is the probability of events. In MCL, the importance factors indicate the matching probabilities between observations and the current sample set. Therefore, we can represent the uncertainty of the distribution of weight values of sample set by entropy.

Effective sample size (ESS) of a weighted sample set is computed by (Liu, 2001):

$$ESS = \frac{N_k}{1 + cv^2} \tag{5}$$

where N_k denotes the sample size at time index *k*, and cv^2 denotes variation of the weight values of samples, derived by

$$cv^2 = \frac{\mathrm{var}(w(i))}{E^2(w(i))} = \frac{1}{N_k} \sum_{1}^{N_k} (N_k \cdot w(i) - 1)^2 \tag{6}$$

in which $E(w(i))$ and $\mathrm{var}(w(i))$ denote the mean and variance of the distribution of weight values of samples, respectively.

If the effective sample size is lower than a given threshold (percentage of the sample size), over-convergence phenomenon is confirmed. It is then necessary to introduce new samples,

with the number of $n_s = c(N_k - ESS)$, where c is a constant. Otherwise, the difference of entropy of the distribution of weight values before and after sensor update is further examined to determine whether the over-convergence phenomenon happens, in the following way

$$\frac{H_c - H_p}{H_p} \geq \lambda \tag{7}$$

where, H_p and H_c denote the entropy of the distribution of weight values before and after sensor update, respectively; $\lambda \in (0,1)$ is a benchmark to check the relative change of entropy, which decreases as H_p increases. The larger the difference is, the more likely over-convergence occurs. When over-convergence is confirmed in this manner, the number of new samples to be introduced is $n_s = (1-\lambda)(N_k - ESS)$.

By the analysis of the distribution of weight values of sample set, the abnormity cases can be effectively checked through the two validation mechanisms, and thereby premature convergence and deviation problem caused by non-modeled events can be deliberately prevented. In addition, more real-time requirements can be satisfied with smaller sample size. Further, the strategy of employing different resampling processes is to construct the true posterior distribution by treating the observation model as part of the proposal distribution, which is guaranteed to be consistent with the observations even when using smaller sample size or more precise sensors.

4. An Implementation of Visual-Based Extended Monte Carlo Localization

In this section, an implementation of the proposed extended MCL algorithm with visual technology will be discussed. The observation model $p(Z_k \mid X_k^{(i)})$ is constructed based on visual polyhedron features that are recognized by Bayesian networks. The triangulation-based resampling is applied.

4.1 Sample Update

In the prediction process, samples are extracted from the motion equation

$$X_k = \mathrm{f}(X_{k-1}, u_{k-1}, v_{k-1})$$

where v_{k-1} denotes the sensor noise during the motion. Note that u_{k-1} consists of the translation Δs_{k-1} and the rotation $\Delta\theta_{k-1}$, which are independent between each other and can be modeled with the odometry model (Rekleitis, 2003b).

When the robot rotates by an angle of $\Delta\theta_{k-1}$, the noise caused by odometry error is modeled as a Gaussian with mean zero and sigma proportional to $\Delta\theta_{k-1}$. Therefore, the heading angle of the robot is updated by

$$\theta_k = \theta_{k-1} + \Delta\theta_{k-1} + \varepsilon_{\Delta\theta_{k-1}} \tag{8}$$

where $\varepsilon_{\Delta\theta_{k-1}}$ is a random noise derived from the heading error model $N(0,\sigma_{rot}\Delta\theta_{k-1})$, and σ_{rot} is a scale factor obtained experimentally (Rekleitis, 2003a). Likewise, there exists a translation error denoted by $\varepsilon_{\Delta s_{k-1}}$, which is related to the forward translation Δs_{k-1}. Furthermore, the change in orientation during the forward translation leads to the heading deviation. Then, the pose of samples can be updated by

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \mathrm{f}(X_{k-1}, u_{k-1}, v_{k-1}) = \begin{bmatrix} x_{k-1} + (\Delta s_{k-1} + \varepsilon_{\Delta s_{k-1}})\cos(\theta_k) \\ y_{k-1} + (\Delta s_{k-1} + \varepsilon_{\Delta s_{k-1}})\sin(\theta_k) \\ \theta_{k-1} + \Delta\theta_{k-1} + \varepsilon_{\Delta\theta_{k-1}} + \varepsilon_{\theta_1} \end{bmatrix} \tag{9}$$

where, $\varepsilon_{\Delta s_{k-1}}$ and ε_{θ_1} are random noises from the error models $N(0,\sigma_{trans}\Delta s_{k-1})$ and $N(0,\sigma_{drift}\Delta s_{k-1})$, σ_{trans} and σ_{drift} are scale factors experimentally obtained for the sigma of these Gaussian models (Rekleitis, 2003a); the sensor noise v_{k-1} includes random noise $\varepsilon_{\Delta\theta_{k-1}}$ estimated by the heading error model $N(0,\sigma_{rot}\Delta\theta_{k-1})$, as well as the translational error $\varepsilon_{\Delta s_{k-1}}$ with Gaussian model of $N(0,\sigma_{trans}\Delta s_{k-1})$ and the heading deviation ε_{θ_1} with zero mean, estimated by $N(0,\sigma_{drift}\Delta s_{k-1})$.

To generate samples, the robot heading angle is firstly calculated by (8), and then the robot pose by (9). Figure 1 illustrates a distribution of samples generated in travelling 3.5 m along a straight line, with a known initial pose (on the right end) and the two noise parameters $(\sigma_{trans},\sigma_{drift})$, where only the two-dimensional pose in x and y directions are given. As shown in this figure, the sample distribution spreads more widely as the travelled distance increases (the solid line with an arrow depicts the odometry data).

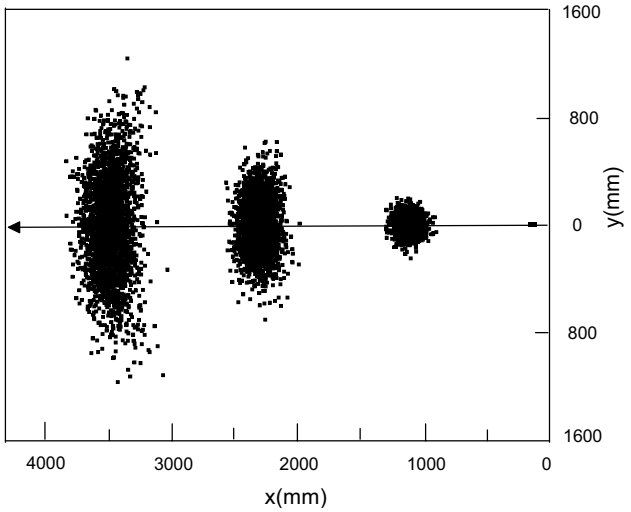


Figure 1. Sample distribution of straight line motion with error $\sigma_{trans} = 5$ and $\sigma_{drift} = 1$

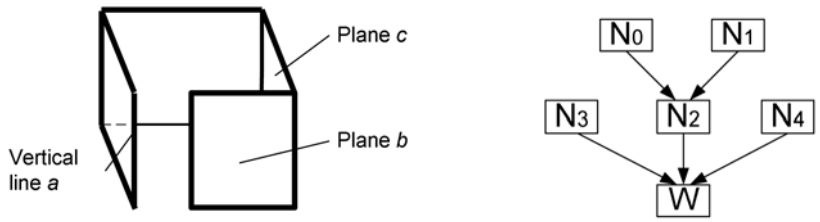
4.2 Visual Sensor Update

Observations from exteroceptive sensors are used to re-weight the samples extracted from the proposal distribution. Observations are based on sensing of polyhedrons in indoor office environments. Using the observed features, an observation model can be constructed for samples re-weighting, and the triangulation-based resampling process can be applied.

Visual polyhedron features

Polyhedrons such as compartments, refrigerators and doors in office environments, are used as visual features in this application. These features are recognized by Bayesian network (Sarkar & Boyer, 1993) that combines perceptual organization and color model. Low-level geometrical features such as points and lines, are grouped by perceptual organization to form meaningful high-level features such as rectangular and tri-lines passing a common point. HIS (Hue, Intensity, Saturation) color model is employed to recognize color feature of polyhedrons. Figure 2 illustrates a model of compartment and the corresponding Bayesian network for recognition. More details about nodes in the Bayesian network can be found in the paper (Shang et al., 2004).

This recognition method is suitable for different environment conditions (e.g., different illuminations and occlusions) with different threshold settings. False-positives and false-negatives can also be reduced thanks to considering polyhedrons as features. Furthermore, there are many low-level features in a feature group belonging to the same polyhedron, which are helpful in matching between observations and environment model since the search area is constrained.



(a) (b)

Figure 2. Compartment model (a) and Bayesian network for compartment recognition (b)

Consider a set of visual features $Z_k = [z_k^1, z_k^2 \cdots, z_k^m]^T$ to be observed. The eigenvector of each visual feature *j*, denoted by $z_k^j = [t_k^j, \phi_k^j]^T$, is composed of the feature type t_k^j and the visual angle ϕ_k^j relative to the camera system, developed by

$$\phi_k^j = \arctan((width/2 - u_k^{cj}) \times \tan(\beta/2) / (width/2))$$

where $width$ is the image width, u_k^{cj} is the horizontal position of feature in the image, and β is half of the horizontal visual angle of the camera system.

Visual observation model

As described in Section 2, the sample weight is updated through an observation model $p(Z_k \mid X_k^{(i)})$. It is assumed that the features are detected solely depending on the robot's pose. Therefore, the observation model can be specified as:

$$p(Z_k \mid X_k^{(i)}) = p(z_k^1, \cdots, z_k^m \mid X_k^{(i)}) = p(z_k^1 \mid X_k^{(i)}) \times \cdots \times p(z_k^m \mid X_k^{(i)}) \tag{10}$$

The observation model for each specific feature j can be constructed based on matching of the feature type and the deviation of the visual angle, i.e.,

$$p(z_k^j \mid X_k^{(i)}) = \delta(t_k^j - t_k^{j0}) \cdot \frac{e^{-\frac{(\phi_k^j - \phi_k^{j0})^2}{2\sigma_\phi^2}}}{\sqrt{2\pi}\sigma_\phi} \tag{11}$$

where, t_k^j and t_k^{j0} are feature types of current and predictive observations, respectively, and $\delta(\cdot)$ is a Dirac function; ϕ_k^j and ϕ_k^{j0} are visual angles of current and predictive observations, and σ_ϕ is variance of ϕ. When $t_k^j = t_k^{j0}$, the observed feature type is the same as the predictive ones. When the number of the predictive features is more than that of the observed ones, only part of the predictive features with the same number of observed features are extracted after they are sorted by the visual angle, and then a maximum likelihood is applied.

4.3 Resampling Step

As discussed in Section 3, two validation mechanisms in the resampling step are firstly applied to check abnormity of the distribution of weight values of sample sets. Then according to the validation, the strategy of using different resampling processes is employed, where samples are extracted either from importance resampling or from observation model. Importance resampling has been illustrated in Section 2 (see Table 1). Here we will discuss the resampling method from the visual observation model.

As we have mentioned that the threshold W_{th} for uniformity validation should be appropriately selected. For our application example, the threshold W_{th} is determined as follows based on the observation model (11):

$$W_{th} = k_w \cdot \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma_\phi} \tag{12}$$

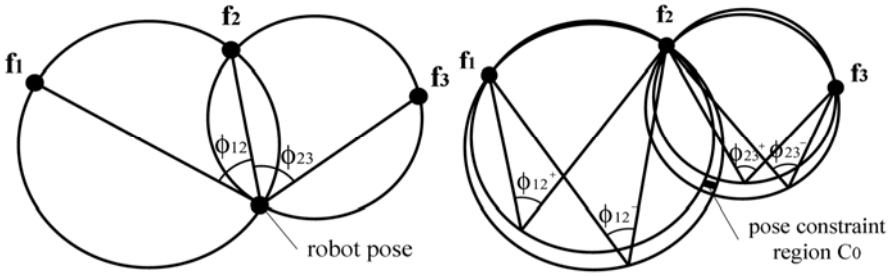
where k_w is a scale factor, and m is the number of current features.

Sensor-based Resampling

In the resampling from observations, $p(X_k \mid Z_k)$ is also treated as the proposal distribution, which can provide consistent samples with observations to form the true posterior distribution. According to the SIR algorithm, samples must be properly weighted in order to represent the probability distribution. Note that such sensor-based resampling is mainly applied in some abnormal cases (e.g., non-modeled events), which is not carried out in every iterative cycle. Furthermore, after completing the sensor-based resampling, all samples are supposed to be uniformly distributed and re-weighted by motion/sensor updates in the next cycle.

The triangulation method is utilized for resampling from visual features, where visual angles are served as the observation features in the application (Krotkov, 1989; Mufioz &

Gonzalez, 1998; Yuen & MacDonald, 2005). Ideally, the robot can be uniquely localized with at least three features, as shown in Figure 3 (a), where the number 1~3 denotes the index of features. In practice, however, there exists uncertainty in the pose estimation due to observation errors and image processing uncertainty. The pose constrained region C_0 shown in Figure 3 (b), illustrates the uncertain area of the robot pose with uncertain visual angle, where $\phi^- = \phi - \Delta\phi$, $\phi^+ = \phi + \Delta\phi$, ϕ and $\Delta\phi$ are visual angle and the uncertainty, respectively. The uncertain pose region just provides a space for resampling. The incorrect samples can be gradually excluded as the update process goes on.



(a) (b)

Figure 3. Triangulation-based localization (a) in the ideal case, (b) in the case with visual angle error

In the existing triangulation methods, visual features are usually limited to vertical edges (Mouaddib & Marhic, 2000), which are quite similar and have large numbers. While in our application, polyhedron features recognized by Bayesian networks combine perceptual organization and color model, and therefore reduce the number of features and simplify the search. In addition, the sub-features of polyhedrons such as vertical edges in recognized compartments, can also be used for triangulation.

In the process of searching features by interpretation tree (IT), the following optimizations can be applied:

1. Consider all polyhedrons as a whole, and the position of each polyhedron as the central position of each individual feature.
2. As visual angle of the camera system is limited, form the feature groups that consist of several adjacent features according to their space layout. The number of features in each feature group should be more than that of the observed features. The search area of the interpretation tree is within each feature group.
3. Search match in terms of the feature type, and then verify by triangulation method the features satisfying the type validation, to see whether the pose constraint regions each formed by visual angles of two features, are intersected as is shown by the pose constraint region C_0 in Figure 3 (b).

Then, the random samples $(x_k^{(i)}, y_k^{(i)})$ can be extracted from the pose constraint region. The orientations of the samples are given by

$$\theta_k^{(i)} = \frac{1}{m}\sum_{j=1}^{m}(\arctan((y_k^{(i)} - y_k^j)/(x_k^{(i)} - x_k^j)) - \phi_k^j) \tag{13}$$

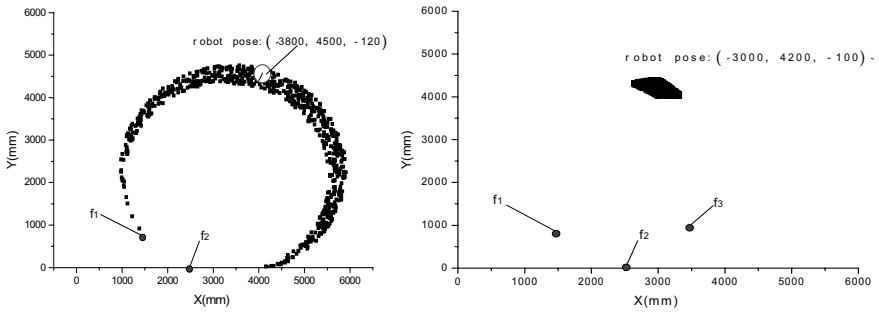
where x_k^j, y_k^j, ϕ_k^j are position and visual angle of the feature *j*, respectively, and m is the number of observed features.

Figure 4 (a) illustrates the sample distribution after sampling from two observed features f_1 and f_2, for a robot pose (3800mm, 4500mm, -120°). The visual angle error is about five percent of the visual angle. There are about 1000 generated samples that are sparsely distributed in the intersection region formed by the observed features. Figure 4 (b) illustrates the sampling results from three features f_1, f_2 and f_3, for a robot pose (3000mm, 4200mm, -100°). It can be seen that all extracted samples locate in the pose constraint region and are close to the true robot pose.

5. Experiments

To verify the proposed extended MCL method, experiments were carried out on a pioneer 2/DX mobile robot with a color CCD camera and sixteen sonar sensors, as shown in Figure 5. The camera has a maximum view angle of 48.8 degrees, used for image acquisition and feature recognition. Sonar sensors are mainly for collision avoidance. Experiments were performed in a general indoor office environment as shown in Figure 6 (a). Features in this environment are compartments (diagonal shadow), refrigerators (crossed shadow) and door (short thick line below). Layout of features is shown in Figure 6 (b).

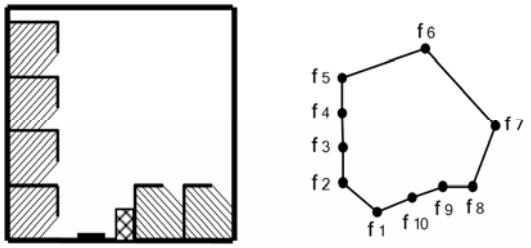
In the experiments, the sample size was set as a constant of 400. Parameters of the extended MCL are: the percentage threshold of effective sample size was 10%, the constant *c* was 0.8, λ was 0.15~0.25, and the scale factor k_w was 50%. Parameters for conventional MCL (with random resampling) were: percentage threshold of effective sample size is still 10%, *c*=0.3, λ =0.35, and k_w =30%.



(a) two features observed (b) three features observed

Figure 4. Sampling from observations with two and three features respectively

Figure 5. Pioneer 2/DX mobile robot, equipped with a color CCD camera and sixteen sonar sensors around



(a) (b)

Figure 6. (a) Experiment environmental model. (b) Layout of environmental features

5.1 Global Localization with Kidnapped Movement

First, the proposed extended MCL method was applied for global localization as well as an non-modeled movement (e.g., kidnapped problem), with time-variant sample distributions, entropy and effective sample size. Figure 7 illustrates the whole motion trajectory. It is seen that the robot started the motion from position *a* to position *b*, then was kidnapped to position *c*, and then continued to move to the end position *d*. Figures 8 and 9 illustrate results when the sample size was within 400. Figure 8 shows that the effective sample size and entropy are time varying. Sample distributions at different iterations are shown in Figure 9, where (a) ~ (f) corresponds to the initial, the 1st, the 8th, the 9th, the 17th, the 18th and the 26th iterations, respectively. As shown in Figure 9 (a), the initial distribution was uniform. At the first iteration, when two compartments f_4 and f_5 were observed, entropy after sensor update decreased, as shown in Figure 8 (b), and both importance resampling and triangulation-based resampling were applied. Due to existence of multi-matches, importance resampling was applied in all successive iterations, as multi-clusters shown in Figure 9 (b). At the 9th iteration, the feature of the door f_1 was observed, and a single sample cluster shown in Figure 9 (c) was obtained until reaching position *b*, where all samples were distributed nearby the true pose of the robot, as seen in Figure 9 (d). At the 18th iteration, the robot was kidnapped to the position *c* with a largely-changed heading angle. At this moment, the effective sample size and the entropy decreased greatly, as shown in Figure 8 (a) and (b). With observations of the features of the door f_1, the refrigerator f_{10}, and the compartment f_9, sample distribution was obtained as shown in Figure 9 (e), after applying importance resampling and triangulation-based resampling, until to the end position *d* where the sample distribution is shown in Figure 9 (f).

From the above localization process, it can be seen that the sensor-based resampling, after an effective examination of weight values of samples by over-convergence validation, can well solve the robot kidnapped problem. Due to too many similar features in the environment, there were still some samples with higher weights after the kidnapped motion, and therefore only over-convergence validation was executed without the uniformity validation. If the robot is kidnapped to a region without similar features, the uniformity validation can be executed.

5.2 Comparison of Localization Errors

Through applying the strategy of different resampling processes in the extended MCL, the localization error becomes smaller than that with the conventional MCL, especially in the

non-modeled movements. Figure 10 illustrates a comparison of the localization errors between the extended MCL and the conventional MCL appended by random resampling, with the same observation model and the same sample size before and after the kidnapped motion. Suppose that the robot's pose obtained from odometry was accurate enough for a short moving distance on the smooth floor. As is seen from Figure 10, at the moment when localization error increases at the 5th iteration, robot was kidnapped. The localization error under the extended MCL (with triangulation-based resampling), is much smaller than that under conventional MCL (with random resampling). This is to verify the improved localization performance of the proposed extended MCL.

5.3 Time Performance

We further demonstrate that the computational resources of the extended MCL could be effectively utilized by appropriately using the sensor-based resampling. As seen from the previous experiments, the number of samples (sample size) with EMCL is only 400, while the number of sample size with conventional MCL is usually much higher than it to obtain good localization performance.

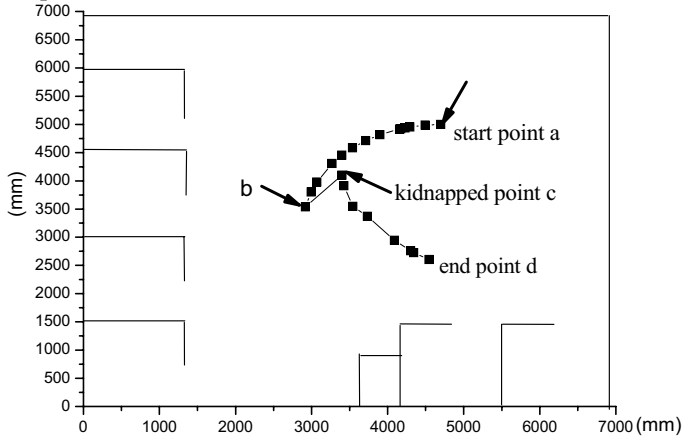


Figure 7. Motion trajectory in localization process (a bit enlarged relative to Figure 6 (a))

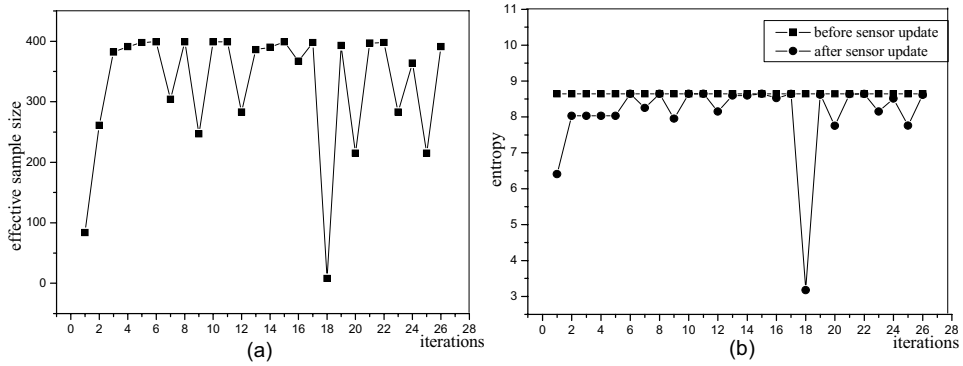


Figure 8. (a) Effective sample size in localization process. (b) Entropy before and after sensor update

Figure 11 (a) illustrates the total update time of the resampling process with respect to different numbers of samples. It can be seen when the sample size is less than 1500, the update time is lower and increases slowly as the sample size increases; when the sample size is more than 2000, the update time is higher and increases fast as the sample size increases. This indicates higher computational efficiency since high number of samples is not required, and thus the update time can be saved.

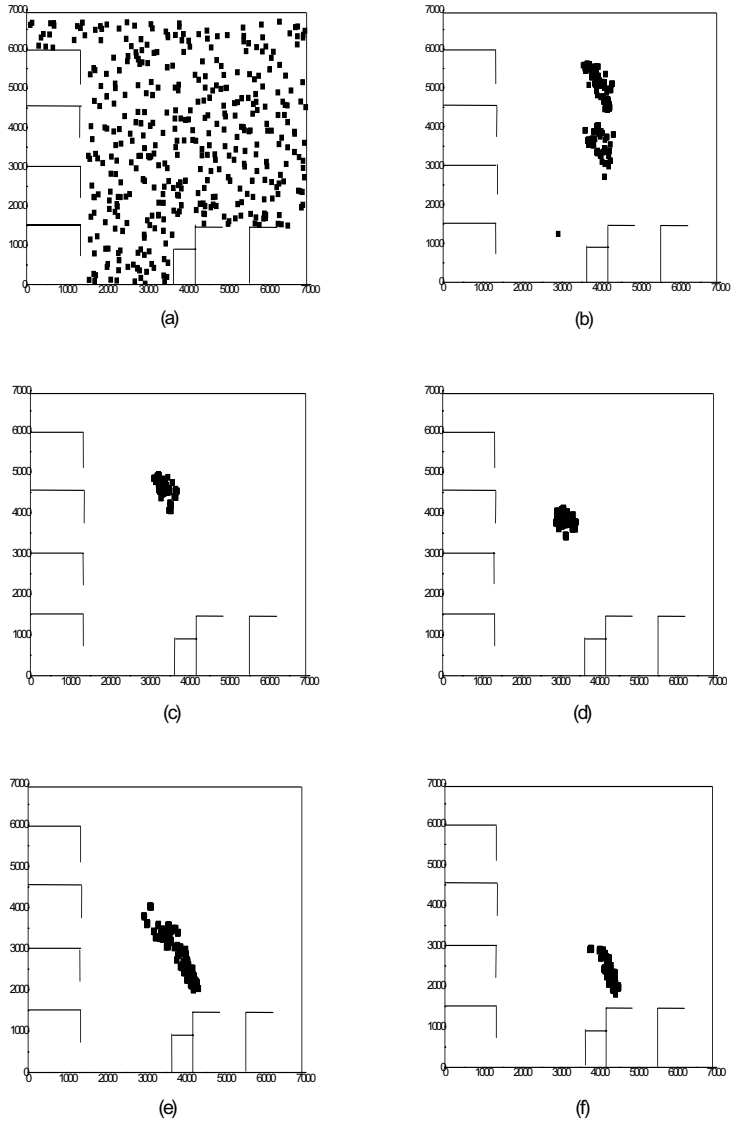


Figure 9. Sample distributions at different iterations (a) initial time and (b) ~ (f) 8th, 9th, 17th, 18th and 26th iterations

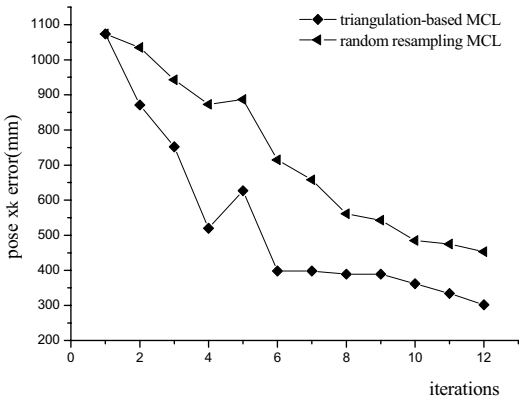
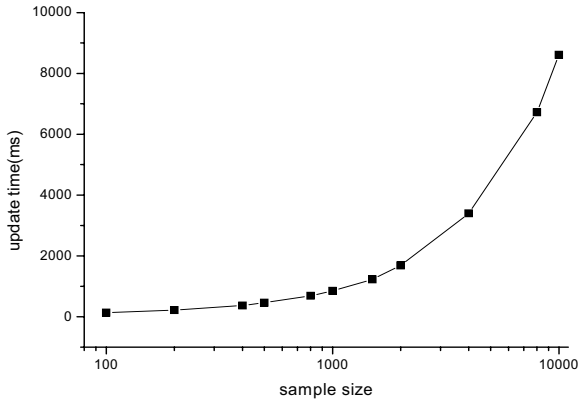
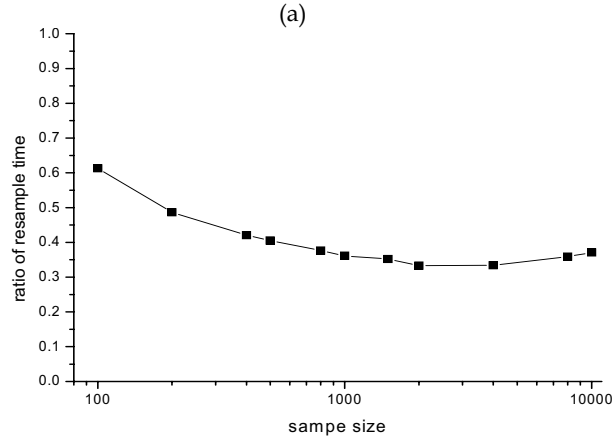


Figure 10. Localization error comparison between extended MCL and MC



(a)



(b)

Figure 11. (a) Update time (b) Percentage of sensor-based resampling time with respect to the total update time

Figure 11 (b) further illustrates the percentage of the sensor-based resampling time with respect to the total update time versus different number of samples. The percentage of resampling time decreases as the sample size increases, i.e., from 48.6% with 200 samples to 35.9% with 8000 samples. Since many samples are not deleted in a large sample set, the sensor-based resampling is not necessarily performed, and the process without resampling dominates the whole process. When the sample size increases to a certain extent, the percentage of the resampling time does not change obviously. This implies that the extended MCL with smaller sample size has the similar localization performance to that with relative larger sample size. Although the percentage of the sensor-based resampling time in the whole update time with smaller sample size is higher than that with higher sample size, the total update time is reduced when using smaller sample size.

6. Conclusion

An extended Monte Carlo localization (EMCL) method is proposed in this book chapter by introducing two validation mechanisms to apply a resampling strategy to conventional MCL. Two validation mechanisms, uniformity validation and over-convergence validation, are effectively used to check abnormity of the distribution of weight values of sample set, e.g., observation deviation or over-convergence problem. The strategy of employing different resampling processes is proposed to construct more consistent posterior distribution with observations. This new approach is aimed to improve localization performance particularly with smaller sample size in the non-modeled robot movements, and thus achieve global localization more efficiently. A vision-based extended MCL is further implemented, utilizing triangulation-based resampling from visual features in a constraint region of the pose space. Experiments conducted on a mobile robot with a color CCD camera and sixteen sonar sensors verify efficiency of the extended MCL method.

7. Acknowledgement

This work was supported in part by a grant from Research Grants Council of the Hong Kong Special Administrative Region, China [Reference No. CityU 119705], and a grant from City University of Hong Kong (project no. 7002127).

8. References

Borenstein, J.; Everett, H. & Feng, L. (1996). Where am I? *Sensors and Methods for Autonomous Mobile Robot Positioning.* Wellesley. Mass.:AK Peters,

Carpenter, J.; Clifford, P. & Fearnhead, P. (1999). Improved particle filter for nonlinear problems, *IEE proceedings on Sonar and Navigation*, pp. 2-7.

Castellanos, J. & Tardos, J. (1996). Laser-based segmentation and localization for a mobile robot, *Proceedings of the 6th Symposium (ISRAM)*, pp. 101-108, ASME Press, New York, NY.

Chenavier, F. & Crowley, J. (1992). Position estimation for a mobile robot using vision and odometry, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2588-2593.

Dellaert, F.; Burgard, W. & Fox, D. (1999). Using the condensation algorithm for robust, vision-based mobile robot localization, *Proceedings of the IEEE International Conference on CVPR*, pp. 588-594.

Doucet, A. (1998). On sequential simulation-based methods for Bayesian filtering, Cambridge University, Department of Engineering, Cambridge, UK, *Technical Report*. CUED/FINFENG/ TR 310.

Drumheller, M. (1987). Mobile robot localization using sonar, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.PAMI-9, No.2, 325–332.

Fox, D. (2003). Adapting the sample size in particle filter through KLD-Sampling, *International Journal of Robotics Research*.

Fox, D.; Burgard, W. & Dellaert, F. (1999a). Monte Carlo localization: efficient position estimation for mobile robots, *Proceedings of the AAAI-9*, pp. 343-3499, Orlando, Florida.

Fox, D.; Burgard, W. & Thrun, S. (1999b). Markov localization for mobile robots in dynamic environments, *Journal of Artificial Intelligence Research*, Vol. 11, 391-427.

Gaspar, J.; Winters, N. & Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omnidirectional camera, *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 6, 890-898.

Luo, R. & Hong, B. (2004). Coevolution based adaptive Monte Carlo localization (CEAMCL), *International Journal of Advanced Robotic Systems*. Vol. 1, No. 3, 183-190.

Isard, M. & Blake, A. (1998). Condensation-conditional density propagation for visual tracking, *International Journal of Computer Vision*, Vol. 29, No. 1.

Jensfelt, P. & Kristensen, S. (2001). Active global localization for a mobile robot using multiple hypothesis tracking, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 5, 748-760.

Jensfelt, P.; Wijk, O.; Austin, D. & Anderson, M. (2000). Experiments on augmenting condensation for mobile robot localization, *Proceedings of the IEEE International Conference on Robotics and Automation*.

Konolige, K. (2001). Robot motion: probabilistic model; sampling and Gaussian implementations; Markov localization, AI Center, SRI International, *Technical Note*.

Kortenkamp, D.; Bonasso, R. & Murphy, R. (1998). (Eds.), *AI-based Mobile Robots: Case Studies of Successful Robot Systems*, MIT Press, Cambridge, MA.

Kraetzschmar, G. & Enderle, S. (2002). Self-localization using sporadic features, *Robotics and Autonomous Systems*, Vol. 40, 111-119.

Krotkov, E. (1989). Mobile robot localization using a single image, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 978-983.

Lenser, S. & Veloso, M. (2000). Sensor resetting localization for poorly modeled mobile robots, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1225-1232.

Leonard, J. & Durrant-White, H. (1991). Mobile robot localization by tracking geometric beacons, *IEEE Transactions on Robotics and Automation*, Vol. 7, 89–97.

Liu, J.; Chen, R. & Logvinenko, T. (2001). A theoretical framework for sequential importance sampling and resampling, In *Sequential Monte Carlo in Practice*, Springer-Verlag, New York,.

Mouaddib, E. & Marhic, B. (2000). Geometrical matching for mobile robot localization, *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 5, 542-552.

Mufioz, A. & Gonzalez, J. (1998). Two-dimensional landmark-based position estimation from a single image, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3709-3714.

Nourbakhsh, I.; Powers, R. & Birchfield, S. (1995). Dervish: An office-navigating robot, *AI Magazine*, Vol. 16, No. 2, 53-60.

Rekleitis, I. (2003a). Cooperative Localization and Multi-Robot Exploration, *PhD Thesis*, School of Computer Science, McGill Univ., Montreal, Quebec, Canada.

Rekleitis, I. (2003b). Probabilistic cooperative localization and mapping in practice, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1907-1912.

Rofer, T. & Jungel, M. (2003). Vision-based fast and reactive Monte-Carlo localization, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 856-861.

Sarkar S. & Boyer, K. L. (1993). Integration, inference, and management of spatial information using Bayesian networks: perceptual organization, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 3, 256-273.

Shang, W.; Ma, X. & Dai, X. (2004). 3D objects detection with Bayesian Networks for vision-guided mobile robot navigation, *Proceedings of the IEEE International Conference on Control, Automation, Robotics and Vision*, pp.1134-1139.

Tardos, J.; Neira, J.; Newman, P. & Leonard, J. (2002). Robust mapping and localization in indoor environments using sonar data, *International Journal of Robotics Research*, Vol. 21, No. 4, 311-330.

Thrun, S.; Fox, D.; Burgard, W. & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots, *Artificial Intelligence*, Vol. 128, 99-141.

Wijk, O. & Christensen, H. (2000). Triangulation based fusion of sonar data with application in robot pose tracking, *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 6, 740-752.

Yuen, D. & MacDonald, B. (2005). Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison, *IEEE Transactions on Robotics*, Vol. 21, No. 2, 217-226.

Vision Systems: Applications

Edited by Goro Obinata and Ashish Dutta

ISBN 978-3-902613-01-1

Hard cover, 608 pages

Publisher I-Tech Education and Publishing

Published online 01, June, 2007

Published in print edition June, 2007

Computer Vision is the most important key in developing autonomous navigation systems for interaction with the environment. It also leads us to marvel at the functioning of our own vision system. In this book we have collected the latest applications of vision research from around the world. It contains both the conventional research areas like mobile robot navigation and map building, and more recent applications such as, micro vision, etc.The fist seven chapters contain the newer applications of vision like micro vision, grasping using vision, behavior based perception, inspection of railways and humanitarian demining. The later chapters deal with applications of vision in mobile robot navigation, camera calibration, object detection in vision search, map building, etc.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821