Experiments on Augmenting Condensation for Mobile Robot Localization

Patric Jensfelt patric@s3.kth.se
Olle Wijk olle@s3.kth.se
David J. Austin d.austin@computer.org
Magnus Andersson sungam@nada.kth.se

Centre for Autonomous Systems,
Royal Institute of Technology,
Stockholm SE-100 44, Sweden.

Abstract

In this paper we study some modifications of the CONDENSATION *algorithm. The case studied is feature based mobile robot localization in a large scale environment. The required sample set size for making the* CONDENSATION *algorithm converge properly can in many cases require too much computation. This is often the case when observing features in symmetric environments like for instance doors in long corridors. In such areas a large sample set is required to resolve the generated multi-hypotheses problem. To manage with a sample set size which in the normal case would cause the* CONDENSATION *algorithm to break down, we study two modifications. The first strategy, called "*CONDENSATION *with random sampling", takes part of the sample set and spreads it randomly over the environment the robot operates in. The second strategy, called "*CONDENSATION *with planned sampling", places part of the sample set at planned positions based on the detected features. From the experiments we conclude that the second strategy is the best and can reduce the sample set size by at least a factor of 40.*

1 Introduction

Mobile robot localization is a field that has attracted much interest from researchers the past few years. For a long time the dominating approach was to approximate the probability density function (PDF) of the robot position with a uni-modal Gaussian distribution which could be propagated using a standard Kalman filter [1, 8]. In most cases these methods were used for robot pose tracking. In recent years methods have been applied that make use of multi-modal representation of the PDF, to enable global robot localization in a large-scale environment. A wide range of different world representations have been used when working with a multi-modal PDF. Nourbakhsh used a topological model for the world [4] and in [9] a grid based representation is used. Recently the CONDENSATION algorithm [5], which use a sample based representation of the PDF, has become popular [2]. The method shows convergence properties when using a large enough sample set. The necessary sample set size depends on the application, and for certain cases the required sample set size can be too large with respect to computational complexity. In this paper we study CONDENSATION applied to feature based localization. The experiments are carried out in a large scale environment, with areas containing feature symmetries, e.g. corridors. In such areas we conclude from the experiments that the CONDENSATION algorithm has unacceptable performance when using a sample set size which is on the limit of what could be handled on a 450MHz PC. To be able to cope with a smaller sample set size, we study two strategies for modifying the CONDENSATION algorithm. The first strategy is not to wait for the sample set to diffuse to new interesting locations, but instead take part of the sample set and spread it randomly over the area the robot operates in. In this way new modes in the PDF are introduced faster and can prevent the algorithm from converging to a false location. However, too much random sampling will prevent the CONDENSATION algorithm from converging at all. The second strategy is to take part of the sample set and spread it at planned positions based on associating observed features to a world map of features.

The paper is outlined as follows. First the notation used in the paper is explained and the assumptions we make for doing feature based localizations are stated.

In section 3 the CONDENSATION algorithm is recapitulated for convenience and in section 4 and 5 the CONDENSATION algorithm is augmented with *random sampling* and *planned sampling* respectively. The paper ends with an experimental section comparing the different versions of the CONDENSATION algorithm on a common data set taken from a large scale environment.

2 Problem statement and assumptions

In the mobile robot localization problem stated here, a (W)orld and a (R)obot coordinate system are present (figure 1). The task is to find the robot location $(x^{(W)}, y^{(W)})$ in world coordinates and the orientation difference α between the two coordinate systems. The following assumptions are made to deal with this problem:

- The robot can measure its position $(x^{(R)}, y^{(R)})$ in robot coordinates using odometry.
- The robot can make external measurements in robot coordinates of features from the surrounding world. The set of features measured in the time interval $[t, t+T]$[1] is denoted $\mathcal{F}$ and the size of $\mathcal{F}$ is denoted $|\mathcal{F}|$. The set $\mathcal{F}$ can be divided into K disjoint subsets $\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, \ldots, \mathcal{F}^{(K)}$ corresponding to K different feature types. Hence $\mathcal{F}$ can be written

$$\begin{aligned} \mathcal{F} &= \mathcal{F}^{(1)} \cup \mathcal{F}^{(2)} \cup \ldots \cup \mathcal{F}^{(K)} \\ \mathcal{F}^{(k)} &= \left\{ f_1^k, f_2^k, \ldots, f_{|\mathcal{F}^{(k)}|}^k \right\} \quad k = 1, \ldots, K \end{aligned}$$

  In the coming sections the set $\mathcal{F}_t$ will denote an ordered version of $\mathcal{F}$ with respect to the feature time stamps $t_1, t_2, \ldots, t_{|\mathcal{F}|}$, i.e.

$$\begin{aligned} &\mathcal{F}_t = \{\ f_1,\ f_2, \ldots f_{|\mathcal{F}|}\ \} \\ &t \le t_1 < t_2 < \ldots < t_{|\mathcal{F}|} \le t + T \end{aligned}$$

- There is a map $\mathcal{M}$ available in world coordinates that contains the same type of features that can be measured by the robot. Using the same notation as introduced for $\mathcal{F}$ we have that

$$\begin{aligned} \mathcal{M} &= \mathcal{M}^{(1)} \cup \mathcal{M}^{(2)} \cup \ldots \cup \mathcal{M}^{(K)} \\ \mathcal{M}^{(k)} &= \left\{ m_1^k, m_2^k, \ldots, m_{|\mathcal{M}^{(k)}|}^k \right\} \quad k = 1, \ldots, K \end{aligned}$$

  The position information of each map feature $m \in \mathcal{M}$ is stored in a pre-defined world coordinate system that is assumed to be related to the robot coordinate system as described in figure 1.

[1] In the experiments we use $T = 5$ sec.

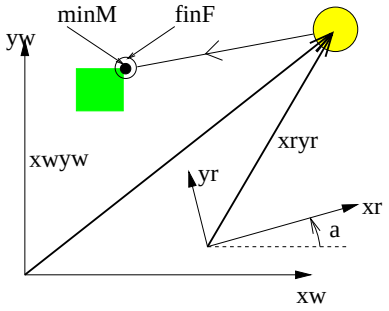

Figure 1: The mobile robot localization problem is here to find the robot pose in world coordinates by using odometry readings $(x^{(R)}, y^{(R)})$ and external feature measurements f that are associated with a world map $\mathcal{M}$.

- There is a sample set

$$S_t = \left\{ (s_1^{(t)}, \pi_1^{(t)}), \ldots, (s_N^{(t)}, \pi_N^{(t)}) \right\},$$

  that represents our distribution of possible robot poses. Here $s_i^{(t)} = (x_i^{(W)}, y_i^{(W)}, \alpha_i)^{(t)}$ suggests that the robot was at position $(x_i^{(W)}, y_i^{(W)})$ at time t and that the orientation difference between the world and robot coordinate system is α_i. Each sample $s_i^{(t)}$ has an attached weight $\pi_i^{(t)}$ that shows its importance relative to the other samples.

3 CONDENSATION

Given the assumptions stated in section 2, the CONDENSATION algorithm could be used for updating the sample set S_t based on feature measurements $f \in \mathcal{F}$ that correspond to a map feature $m \in \mathcal{M}$. The CONDENSATION algorithm is given in pseudo code format in figure 2.
We have modeled the prediction and diffusion term p_d that appears in the CONDENSATION algorithm as

$$p_d = \begin{pmatrix} D(1+D_r)\cos(\alpha_i^{(t_{j-1})} + \beta) \\ D(1+D_r)\sin(\alpha_i^{(t_{j-1})} + \beta) \\ D\alpha_r \end{pmatrix}^T$$

where

$$\begin{aligned} D &= \sqrt{(x_{new}^{(R)} - x_{old}^{(R)})^2 + (y_{new}^{(R)} - y_{old}^{(R)})^2} \\ \beta &= \arctan\left(\frac{y_{new}^{(R)} - y_{old}^{(R)}}{x_{new}^{(R)} - x_{old}^{(R)}}\right) \end{aligned}$$

STEP 1:
Draw samples $s_i^{(t_0)}$ $i = 1, 2, \ldots, N$ based on the current distribution $\pi_1^{(t_0)}, \pi_2^{(t_0)}, \ldots, \pi_N^{(t_0)}$ and put $s_i^{(t_0)} := s_i^{(t_0)}, \quad \pi_i^{(t_0)} := \frac{1}{N}, \quad i = 1, \ldots, N$

STEP 2:
for $j = 1 \rightarrow |\mathcal{F}_t|$ { (cronologic loop)
  for $i = 1 \rightarrow N$ { (loop over all samples)
    Predict and diffuse each sample position:
    $s_i^{(t_j)} := s_i^{(t_{j-1})} + p_d$
    Update the sample weight based on the observed feature f_j:
    $\pi_i^{(t_j)} := \pi_i^{(t_{j-1})} p(f_j | s_i^{(t_j)})$
  }
}

STEP 3:
Put $t_0 := t_{|\mathcal{F}|}$ and wait for new features ($\mathcal{F}_{t+T}$).

Figure 2: The CONDENSATION algorithm applied to feature based localization of a mobile robot.

$$\begin{aligned} D_r &\sim N(0, \sigma_D) \\ \alpha_r &\sim N(0, \sigma_\alpha) \end{aligned}$$

Here $(x_{new}^{(R)}, y_{new}^{(R)})$ is the odometry reading taken at time t_j when feature f_j was detected, and $(x_{old}^{(R)}, y_{old}^{(R)})$ is an odometry reading taken at the time t_{j-1} when feature f_{j-1} was detected. The diffusion parameters D_r and α_r are modeled as independent normally distributed random variables with zero mean and standard deviations σ_D and σ_α. The standard deviations were chosen so that the diffusion will be at least of the same order as the odometry drift of the robot platform. Hence σ_D and σ_α could be determined experimentally for a specific platform[2].
In the update stage of the CONDENSATION algorithm the probability $p(f_j | s_i^{(t_j)})$ needs to be computed. We refer to [6] for an example.
The sample set representation of conditional density used in the CONDENSATION algorithm asymptotically approaches the true distribution as the size N of the sample set size goes to infinity [5, 3]. However, from an implementation point of view, this can be a problem if the computational burden for running the CONDENSATION algorithm becomes too large when using the required sample set size N. One such case that we will study is global robot localization in large scale symmetric environments. Corridors can for instance be classified as symmetric because of their tendency to repeat various kind of features like doors and windows. In such areas the CONDENSATION algorithm will be dependent on a large sample set size N to resolve the multi-hypotheses-problem that is generated by the observed features. If N is too low, the CONDENSATION algorithm is likely to break down, i.e. not converge at all or converge to a false pose hypothesis . In this paper we experimentally investigate how modifications of the CONDENSATION algorithm can improve the convergence properties. In particular, we wish to improve the convergence of CONDENSATION for smaller sample set sizes. We will consider two strategies in the coming sections, CONDENSATION *with random sampling* and CONDENSATION *with planned sampling*.

[2] In our experiments (conducted on a Nomad 200) $\sigma_D = 0.005$ and $\sigma_\alpha = 0.025$ rad/m

4 CONDENSATION with random sampling

When using the CONDENSATION algorithm on a small sample set size N in a large scale environment it is likely that the sample set does not give support where the true PDF has support. As a consequence of this, the time it takes for the sample set to diffuse to the correct location can become unacceptably long. An idea is then to use part of the sample set for initiating new pose hypotheses of the robot. The CONDENSATION algorithm can then be modified as in figure 3. The degree of random sampling is here determined by a parameter $p_r \in [0, 1]$. If $p_r = 1$ then the algorithm will in each iteration spread *all* samples uniformly over the known environment. Such a strategy will of course prevent the algorithm from ever converging and hence a more moderate degree of random sampling should be used. The optimal parameter choice on p_r will be dependent on the kind of features that can be detected. In section 6 we will experimentally evaluate different degrees of random sampling.

5 CONDENSATION with planned sampling

An alternative to random sampling is to put $p_p N$ ($p_p \in [0, 1]$) samples at planned positions by drawing from the PDF $p(\mathbf{x}|\mathcal{F})$, which represent the distribution of probable robot positions when mapping $\mathcal{F}^{(k)}$

STEP 1:
Draw samples $s_i^{'(t_0)}$ $i = (p_r N + 1), \ldots, N$ randomly based on the distribution $\pi_1^{(t_0)}, \pi_2^{(t_0)}, \ldots \pi_N^{(t_0)}$ and put $s_i^{(t_0)} := s_i^{'(t_0)}$.
Draw *completely* new samples $s_i^{(t_0)}$ $i = 1, \ldots, p_r N$ from a uniformly distribution over the environment the robot operates in.
Assign all samples the same weight:
$\pi_i^{(t_0)} = \frac{1}{N} \qquad i = 1, 2, \ldots, N.$

STEP 2 & 3:
Same as in figure 2.

Figure 3: CONDENSATION algorithm augumented with random sampling

to $\mathcal{M}^{(k)}$ $k = 1, 2, \ldots, K$. By assuming independence between the detected features we have

$$p(\mathbf{x}|\mathcal{F}) = \prod_{k=1}^{K} \prod_{j=1}^{|\mathcal{F}^{(k)}|} p(\mathbf{x}|f_j^k) \tag{1}$$

and hence a total of $p_p N$ planned samples are drawn from

$$p(\mathbf{x}|f), \quad f \in \mathcal{F} \tag{2}$$

rather than $p(\mathbf{x}|\mathcal{F})$. A relative quality number[3] q_k is introduced to decide how many planned samples that should be drawn from $p(\mathbf{x}|f_j^k)$. If for instance the feature set $\mathcal{F}$ only holds two features, one of type k_1 and one of type k_2, i.e. $\mathcal{F} = \{f_1^{k_1}, f_1^{k_2}\}$, then $q_{k_1} p_p N/(q_{k_1} + q_{k_2})$ planned samples are drawn from $p(\mathbf{x}|f_1^{k_1})$ and $q_{k_2} p_p N/(q_{k_1} + q_{k_2})$ planned samples are drawn from $p(\mathbf{x}|f_1^{k_2})$. More generally, if there are several features of the same type in $\mathcal{F}$, there are

$$n_p = \frac{q_k}{\sum_{l=1}^{K} q_l |\mathcal{F}^{(l)}|} p_p N \tag{3}$$

planned samples drawn from $p(\mathbf{x}|f^k)$ where $f^k \in \mathcal{F}^{(k)} \subseteq \mathcal{F}$. The CONDENSATION algorithm with planned sampling is given in pseudo code format[4] in figure 4. The degree of planned sampling (p_p) that

[3] A discussion of how to determine the relative quality numbers q_k, $k = 1, 2, \ldots, K$ is found in [6].

[4] This is not an optimal implementation since we are actually predicting, diffusing and performing measurement update on samples which later on are overwritten by planned samples. The pseudo code however becomes easier to read in this manner.

STEP 1:
Same as in figure 2.

STEP 2:
$n_0 := 0$
for $j = 1 \rightarrow |\mathcal{F}_t|$ { (cronologic loop)
  for $i = 1 \rightarrow N$ { (loop over all samples)
    Predict, diffuse and perform measurement update as in figure 2.
  }
  Compute n_p (eq (3)).
  for $i = n_0 + 1 \rightarrow n_0 + n_p$ { (planned sampling)
    Draw a planned sample $s_i^{(t_j)}$ from $p(\mathbf{x}|f_j)$ and give it weight $\frac{1}{N} p(f_j|\mathbf{x})$.
  }
  $n_0 := n_0 + n_p$
}

STEP 3:
Same as in figure 2.

Figure 4: The CONDENSATION algorithm with planned sampling.

should be used is dependent on the kind of features that can be detected. In section 6 the parameter p_p is experimentally determined. It is obvious that a too large degree of planned sampling ($p_p = 1$ extreme case) would make the algorithm "jump" between different pose hypotheses. Another issue is that planned sampling can detract from the CONDENSATION algorithm if many features in $\mathcal{F}$ lack a correspondance in the map $\mathcal{M}$. Such features can arise either from false measurements (outliers) or correct measurements of features that have not been modeled in the map. Of course such feature measurements would also hinder in the original version of the CONDENSATION algorithm or the version using random sampling, but in the planned sampling case the negative effect is more instantaneous.

6 Experiments

The experiments for comparing the algorithms presented in section 3,4 and 5 were performed on two floors in an old hospital building. A map of the building is shown in figure 5. The robot used in the experiments was a Nomad 200 robot equipped with 16 sonars

mounted in a ring and a SICK laser scanner. The sonars were used to detect stable *point features* in the environment by using a method called Triangulation Based Fusion (TBF) [11, 10]. The point features were recorded manually by joysticking the robot around in the building, and subsequently saved in world coordinates in the world map $\mathcal{M}$. The laser scanner was used to detect *line features* as well as *doors*. The line features in the map $\mathcal{M}$ correspond to walls in the rooms and were measured by hand. Since a rectangular model was used for each room, only four wall lines per room were saved in the world map $\mathcal{M}$. The door features were also measured by hand.
Ten data sets were collected with the robot in different parts of the building as a base for the experiments. In figure 5 the ten data sets can be associated with a particular robot trajectory. During the collection of each data set an explore behavior was used to move the robot around in the environment. The behavior had no memory of where the robot had been before and was designed to take the robot to open space areas. To evaluate the performance, the true pose of the robot was also stored in each data set using a well established laser based pose tracker [7] (accuracy ≈5 cm). A search was done for the first iteration where $p_{\text{th}}N$ samples had converged to a blob less than 1 meter in radius. The threshold parameter p_{th} was set according to the formula[5]

$$p_{\text{th}} = 0.9(1 - p_r - p_p), \tag{4}$$

where p_r and p_p are the degree of random and planned sampling respectively. As a success indicator of an experiment it was checked if the "sample blob" corresponded to the "true" robot pose (specified by the laser pose tracker). If this was not the case the experiment was classified as *false converging*. Another failure type was the algorithm *not converging* at all. In the coming subsections no distinction is made between these two failure types.

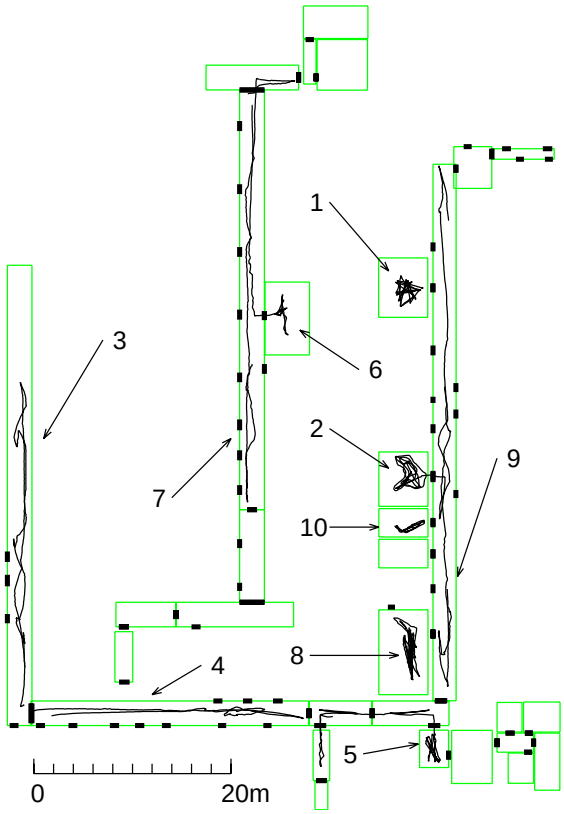

Figure 5: The drawing of the building where the experiments took place. Two floors in the building were used in the experiments. The thick lines are the doors that were used as one of the feature types in the world map $\mathcal{M}$. The robot trajectories show where the experiments took place in the building. Each experiment has got a number attached to it in the figure. The map covers over 200 meters of corridor (with an almost constant width of 2.5 m).

Condensation

The first experiment that was done was to run the original Condensation algorithm (section 3) on the ten data sets varying the sample set size N. Each run over a data set was repeated 10 times to get some statistics to base our conclusions on[6]. The result from the experiment is documented in table 1. From the table it is clear that the Condensation algorithm converges better when increasing the sample set size N. At $N = 100000$ the convergence result is on average 61%. When looking closer to the table it is noted that the corridors corresponding to data sets 3 and 4 (figure 5) were particularly difficult to localize in. The reason for this was the low density of features in these areas and the fact that all doors were closed and hence not detected by the laser scanner. Regarding data set 3, no true convergence was seen in *any* of the experiments presented in this paper[7]. Hence 90% was the maximum convergence result obtained in the experiments.

[5] Random and planned sampling were tested separately in the experiments. Hence either $p_r = 0$ or $p_p = 0$.

[6] Remember that the diffusion part in the Condensation algorithm involves randomization and hence repeated experiments on the same data set will generate different results.

[7] Including some undocumented ones at $N = 200000$.

CONDENSATION								
Exp	$N = 10000$		20000		50000		100000	
1	20%	(21)	40%	(30)	60%	(45)	70%	(33)
2	0%	(-)	60%	(38)	80%	(32)	100%	(35)
3	0%	(-)	0%	(-)	0%	(-)	0%	(-)
4	0%	(-)	20%	(62)	10%	(46)	40%	(71)
5	10%	(44)	30%	(27)	50%	(23)	40%	(27)
6	40%	(33)	10%	(19)	90%	(33)	100%	(31)
7	0%	(-)	10%	(56)	30%	(64)	40%	(67)
8	50%	(35)	90%	(18)	100%	(16)	100%	(14)
9	20%	(34)	40%	(46)	60%	(38)	90%	(26)
10	10%	(42)	0%	(-)	20%	(20)	30%	(18)
Avg	15%	(34)	30%	(33)	50%	(33)	61%	(33)

Table 1: Result when running the CONDENSATION algorithm over varying sample set sizes N. The notation $x\%$ (y) occurring in the table means that $x\%$ of the runs converged to the true position and that the convergence took on average y iterations.

CONDENSATION with random sampling ($N = 50000$)								
Exp	$p_r = 0.01$		0.1		0.2		0.4	
1	20%	(16)	50%	(40)	60%	(38)	100%	(55)
2	90%	(36)	90%	(34)	100%	(34)	100%	(52)
3	0%	(-)	0%	(-)	0%	(-)	0%	(-)
4	20%	(42)	40%	(55)	30%	(57)	40%	(65)
5	30%	(20)	30%	(40)	60%	(36)	100%	(48)
6	80%	(24)	100%	(37)	90%	(59)	0%	(-)
7	30%	(78)	40%	(99)	70%	(82)	20%	(88)
8	100%	(16)	100%	(17)	100%	(16)	100%	(19)
9	60%	(31)	80%	(20)	90%	(43)	90%	(67)
10	20%	(24)	70%	(41)	90%	(39)	80%	(24)
Avg	45%	(30)	60%	(38)	69%	(43)	63%	(48)

Table 2: Result when varying the degree of random sampling p_r when $N = 50000$. The notation in the table is the same as in figure 1. A clear improvement is seen when comparing the best column in this table ($p_r = 0.2$) with the $N = 50000$ column in table 1.

CONDENSATION with random sampling

The second experiment was to run the CONDENSATION algorithm with random sampling. In this experiment we used a fixed sample set size $N = 50000$ and instead varied the degree of random sampling p_r. The results are shown in table 2 and should be compared with the $N = 50000$ column of table 1 (pure CONDENSATION). From the bottom row (average values) of table 2 it is clear that an improved performance can be gained at random sampling degrees $p_r : 0.1 \rightarrow 0.4$. The best result was obtained at $p_r = 0.2$ with an average convergence result of 69%.

CONDENSATION with planned sampling

The third experiment was to run the CONDENSATION algorithm with planned sampling. The sample set size in this experiment was fixed at $N = 10000$ and the degree of planned sampling p_p was varied. The outcome of the experiment is shown in table 3. From

CONDENSATION with planned sampling ($N = 10000$)								
Exp	$p_p = 0.01$		0.02		0.05		0.1	
1	50%	(47)	80%	(32)	100%	(40)	100%	(42)
2	100%	(31)	100%	(27)	100%	(30)	100%	(36)
3	0%	(-)	0%	(-)	0%	(-)	0%	(-)
4	50%	(64)	70%	(64)	90%	(68)	0%	(-)
5	80%	(23)	100%	(28)	100%	(24)	90 %	(27)
6	100%	(29)	100%	(31)	60%	(33)	0%	(-)
7	80%	(93)	100%	(82)	80%	(122)	0 %	(-)
8	100%	(9)	100%	(8)	100%	(10)	100%	(22)
9	100%	(25)	100%	(18)	100%	(19)	100%	(29)
10	70%	(34)	100%	(36)	100%	(23)	100%	(27)
Avg	73%	(37)	85%	(36)	83%	(40)	59%	(31)

Table 3: Result when varying the degree of planned sampling p_p. The notation in the table is the same as in figure 1. The best column in this table ($p_p = 0.02$) is superior to any column in tables 1 and 2 although the sample set size N is only 10000.

the table it is clear that planned sampling is superior to pure CONDENSATION (table 1) and CONDENSATION with random sampling (table 2). The best result was obtained at $p_p = 0.02$ with an average convergence result of 85%. When increasing p_p to 0.1 the algorithm tended to give "jumpy" results and hence there was a decrease in performance.

Comparing the methods

In the fourth experiment the three different versions of the CONDENSATION algorithm were compared by varying the sample set size N from 1000 to 100000. Based on the earlier presented results, the parameter p_r was set to 0.2 when running the CONDENSATION algorithm with random sampling, and p_p to 0.02 when running the CONDENSATION algorithm with planned sampling. A graph comparing the methods is shown in figure 6. From the graph it is evident that random sampling performs better than pure CONDENSATION at all tested sample set sizes N, and that planned sampling performs much better than random sampling. When studying the graph more closely, for instance by fixing the convergence result to a number above 50% and measuring the corresponding sample set sizes for the different curves , it is found that CONDENSATION with random sampling can manage with about $2 - 2.5$ times less sample set size than pure CONDENSATION. For CONDENSATION with planned sampling the corresponding factor is $40 - 50$.

Conclusions

In this paper the CONDENSATION algorithm was augumented with random and planned sampling for

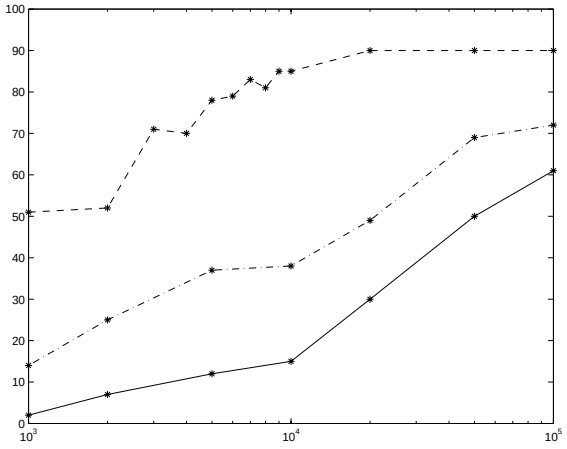

Figure 6: Convergence to true position (%) vs sample set size N. The solid line represents pure CONDENSATION, the dashed-dotted line represents CONDENSATION with random sampling and the dashed line represents CONDENSATION with planned sampling. The maximum level that could be reached was 90% because of that one of the ten data sets contained too few detectable features. Hence no convergence at all was seen on this data set.

mobile robot localization in a large scale environment covered by a world map of features. The features used were doors and lines filtered from a laser scanner and stable point landmarks filtered from sonar data. The outcome of the experiments was that CONDENSATION with random sampling can reduce the sample set size at least 2 times compared to using pure CONDENSATION. When using CONDENSATION with planned sampling the sample set size can be reduced at least a factor 40 compared to pure CONDENSATION.

7 Acknowledgment

This research has been sponsored by the Swedish Foundation for Strategic Research through the Centre for Autonomous Systems. The funding is gratefully acknowledged.

References

[1] J. L. Crowley. World modeling and position estimation for a mobile robot. In *IEEE Intl. Conf. on Robotics and Automation*, volume 3, pages 1574–1579, 1989.

[2] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *IEEE Intl. Conf. on Robotics and Automation*, pages 1322–1328, May 1999.

[3] U. Grenander, Y. Chos, and D. M. Keenan. *A Pattern Theoretical Study of Biological Shapes*. Springer-Verlag, 1991.

[4] I.Nourbakhsh, R. Powers, and S. Birchfield. Dervish an office-navigating robot. *AI Magazine*, 16:53–60, 1995.

[5] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Intl. Journal of Computer Vision*, 29(1):5–28, 1998.

[6] Patric Jensfelt, David Austin, Olle Wijk, and Magnus Andersson. Feature based condensation for mobile robot localization. In *IEEE Intl. Conf. on Robotics and Automation*, 2000.

[7] Patric Jensfelt and Henrik Christensen. Laser based position acquisition and tracking in an indoor environment. In *Proc. of the Intl. Symposium on Robotics and Automation*, volume 1, pages 331–338, Saltillo, Coahuila, Mexico, December 1998. IEEE.

[8] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.

[9] Sebastian Thrun, Arno Bucken, Wolfram Burgard, Dieter Fox, Thorsten Frohlingshaus, Daniel Henning, Thomas Hofmann, Michael Krell, and Timo Schmidt. *Map Learning and High-Speed Navigation in RHINO*, chapter 1, pages 21–54. The MIT Press, 445 Burgess Drive, Menlo Park, CA 94025, 1998.

[10] Olle Wijk and Henrik Christensen. Triangulation based fusion of sonar data for robust robot pose tracking. *submitted to IEEE Transactions on Robotics and Automation*, 1999.

[11] Olle Wijk, Patric Jensfelt, and Henrik Christensen. Triangulation based fusion of ultrasonic sensor data. In *IEEE Intl. Conf. on Robotics and Automation*, volume 4, pages 3419–24, Leuven, Belgium, May 1998.