

Feature Based Condensation for Mobile Robot Localization

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

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

Abstract

Much attention has been given to CONDENSATION methods for mobile robot localization. This has resulted in somewhat of a breakthrough in representing uncertainty for mobile robots. In this paper we use CONDENSATION with planned sampling as a tool for doing feature based global localization in a large and semi-structured environment. This paper presents a comparison of four different feature types: sonar based triangulation points and point pairs, as well as lines and doors extracted using a laser scanner. We show experimental results that highlight the information content of the different features, and point to fruitful combinations. Accuracy, computation time and the ability to narrow down the search space are among the measures used to compare the features. From the comparison of the features, some general guidelines are drawn for determining good feature types.

1 Introduction

The problem dealt with in this paper is the problem of Mobile Robot Localization (MRL), a problem as old as the field of mobile robotics. In short we define the problem as finding the pose¹ of the robot using a map of the environment and sensor data. The map of the environment is not assumed to be complete, in the sense that every single detail is present, but it is assumed that localization is only performed within the area covered by the map.

Being a fundamental problem in mobile robotics, localization is covered by a wide range of literature. We refer to surveys on the topic (e.g. [1]), for a more general discussion of different methods. A brief summation of the history is that uni-modal Kalman fil-

ter based techniques (e.g. [2, 3]) dominated during the 1980's and until recently. These approaches dealt mostly with pose tracking, that is, maintaining an estimate of the pose of the robot. The main assumption of these methods is that the probability density function (PDF) can be approximated with a single Gaussian. In [4] multiple Gaussians are used to represent the PDF, thereby enabling a multi-modal representation at the same time as staying within the framework of the well understood Kalman filter.

This paper focuses on what in [5] is called CONDENSATION and later in [6] Monte Carlo Localization (MCL). The key idea is to use a set of samples to represent the PDF encoding the robots knowledge about its position. In [7] a grid was used to represent the PDF, the drawback of this being the tradeoff between accuracy and computational effort to keep it updated. For large search spaces where the probability density is low over much of the space, a sample based representation is more efficient [6]. For the set of samples to be a good approximation of the true PDF it is important that the samples give support where the PDF differs significantly from zero. In this paper an algorithm from [8] called CONDENSATION with planned sampling is used. In Section 2 the algorithm is described briefly.

In this paper we will discuss the use of different environmental features as landmarks for localization. We will emphasize the problem of understanding what properties a feature should have to be “good” for localization. Hence, we will conduct a comparative study of four different features. In Section 3 some measures are introduced for comparing the features. The different features in the comparison will be presented in Section 4 along with general comments regarding some implementational details, followed by Section 5 where the experimental results are presented. Conclusions are drawn in the final section.

¹By pose we mean position and orientation of the robot in the plane. Position refers to the (x, y) -coordinates of the robot.

2 CONDENSATION

Let the pose of the robot be given by $\mathbf{x} = (x, y, \alpha)^T$, $\mathcal{F}$ the set of detected features and f^k a feature of type k , where $k = 1, \dots, K$. The number of different feature types is K . The task is to determine the PDF $p(\mathbf{x}|\mathcal{F})$, which is represented by a set of samples $\{(s_i, \pi_i)\}$, $i = 1, \dots, N$ when using CONDENSATION. The weight of each individual sample is π_i . The sample set is initialized using prior information about the PDF. When doing global localization, the prior PDF is typically a uniform distribution over the entire known environment.

There are three main steps in the CONDENSATION algorithm that will update the sample set so as to represent the PDF at the next time instant.

1. Re-sampling Phase

Create a new sample set by repeated sampling from the previous set of samples, such that the probability of picking one of the previous samples, s_i , is proportional to its weight, π_i . The new samples are normalized by setting their weights to $\frac{1}{N}$.

2. Predict and Diffusion Phase

In the predict and diffusion phase odometric information is used, along with a model thereof, to determine how the PDF changes as the robot moves. Moving without using feature information has the intuitive result that we lose information, and the PDF will approach a uniform distribution.

3. Update Phase

In the update phase, sensor information (external sensors) is incorporated into the PDF. Let $\mathcal{F}$ be the set of all new features detected during the time interval $[t, t + T]$, $\mathcal{F} = \{f_j^k, j = 1, \dots, |\mathcal{F}^{(k)}|, k = 1, \dots, K\}$, where K is the number of different types of features and $|\mathcal{F}^{(k)}|$ denotes the number of detected features of type k . The update is done by multiplying the weight of each sample, s_i by the probability of observing $\mathcal{F}$ at the position given by s_i ,

$$\pi_i^{(t)} := \pi_i^{(t-T)} \cdot p(\mathcal{F}|s_i).$$

Assuming independence of the features, the probability can be factored into $p(\mathcal{F}|s_i) = p(f_1^1|s_i) \dots p(f_{|\mathcal{F}^{(K)}|}^K|s_i)$.

The basic mechanism is that samples representing poses where $p(\mathcal{F}|s_i)$ is high will be given more weight

and therefore will be re-sampled more often in the **Re-sampling Phase** of the next iteration. Regions where $p(\mathcal{F}|s_i)$ is significant will thus attract many samples, gathering support for a good approximation of the PDF. Having said this, it follows that if there are no samples in an area where the PDF is significant, no samples can be attracted there, and hence the estimate of the PDF in that area will not be good. A special, and crucial case, is when there are no samples close to the true pose of the robot when the sample set is initialized. No matter how many features that are detected, the true pose of the robot will not be found, unless some samples diffuse to the true pose. One simple way to increase the chance of sample support where needed, is to draw some of the samples in the **Re-sampling Phase**, not from the previous sample set but rather from a uniform distribution [8]. However, the probability of success with random sampling is still low in a large environment or when working in a high dimensional space. In [8] this problem is approached by instead drawing some samples from $p(\mathbf{x}|f_j^k)$. This algorithm is called CONDENSATION with planned sampling and is used in the experiments later in this paper. A sample set size of 10000 with 200 planned samples in each iteration is used.

Note that, since the PDF is only updated with detected features, missed detections do not adversely impact the PDF. However, missed detections will slow the convergence. As a result the false positive detections are the most significant error that we must consider.

3 Performance Measures

In this paper we use four measures for comparing the different features.

Computation Time A simple measure is the computational time that is spent on a particular feature, both for detecting it and to use it in updating the PDF (that is to evaluate $p(f_j^k|\mathbf{x})$).

Accuracy The computation time measure alone will only tell us to use simple features, and not anything about what information they provide. There is thus a need for other measures as well, for instance the accuracy that is obtained in localization, using the different features.

Convergence Along with accuracy, the effect of a feature on the convergence is also a possible measure.

That these two last factors might not always point in the same direction is clear from a simple example with two lines forming a corner. Assuming approximate knowledge about the pose of the robot, a very exact estimate can be found using these lines, but if on the other hand the pose is not known at all, there will be many places in a typical indoor environment which will match two lines forming a corner and hence the convergence using lines alone might not be good.

Planned Sample Quality In Section 2 we introduced the idea of planned sampling, that was based on drawing a fraction of the samples directly from the distribution $p(\mathbf{x}|f_j^k)$. The ideal feature, f_j^k , has a $p(\mathbf{x}|f_j^k)$ with support only in very small areas, so that sampling from it will result in a high probability of adding a sample where $p(f_j^k|\mathbf{x})$ is significant. This effect can also be used when evaluating the effectiveness of the feature.

In this paper the relative quality measure, q_k , is based on the probability that a single planned sample will be placed within 0.5 m from the true pose.

4 Implementation

Each feature f_j^k is given access to a world map, $\mathcal{M}^{(k)} = \{m_l^k, l = 1, \dots, |\mathcal{M}^{(k)}|\}$, where m_l^k is a map feature of type k and $|\mathcal{M}^{(k)}|$ is the number of such features in the map. Each f_j^k can evaluate $p(f_j^k|\mathbf{x})$ and $p(\mathbf{x}|f_j^k)$, where $p(f_j^k|\mathbf{x})$ is used in the **Update Phase** and $p(\mathbf{x}|f_j^k)$ to produce planned samples [8]. When combining different features, a relative quality measure, q_k , is given to each feature type. This quality measure is used when determining how many planned samples each feature should be allowed to produce.

No feature detector is perfect and the map will never be complete, therefore $p(f_j^k|\mathbf{x})$ is never zero. The better the detector, the lower the minimum value $\beta_k = \min_{\mathbf{x}} p(f_j^k|\mathbf{x})$ will be. β_k thus incorporates uncertainty both in detecting a feature and the fact that the map is incomplete. β_k is here approximated as a global constant, despite the fact that the map has variable quality. A low β_k corresponds to a feature which can be extracted with very few false positives. It is important that β_k is not made lower than is warranted, as excessively low values might result in the true pose being removed if a false measurement is received. The values β_k are experimentally determined.

The value $1 - \beta_k$ is a measure of the probability that a detected feature actually corresponds to a map feature. This in combination with an “average” feature

of each type is used when calculating q_k .

4.1 Sonar Based Triangulation Points

In [9] and [10] a triangulation based method for ultrasonic sensor data is introduced. In a typical indoor environment, some objects will give rise to very robust responses from the sonar. By using a triangulation technique, these strong response points can be detected and used as map features, $m_l^{TriPoint}$. In the following the triangulation points will simply be referred to as *TriPoints*. Figure 1 illustrates the basic idea of the TBF (Triangulation Based Fusion) algorithm, and Figure 2 shows the map used in the

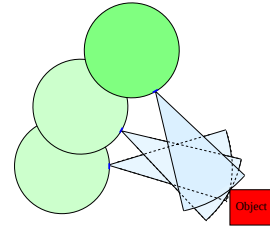


Figure 1: The TBF algorithm associates multiple sonar readings and fuses them into an estimate of the position of the reflecting object.

experiments. The *TriPoint* features are marked with small dots and typically originate from various kinds of vertical edges, e.g. door posts. The *TriPoint* map features $\mathcal{M}^{(TriPoint)}$ represent part of the environmental model, $\mathcal{M}$, that the robot has access to. The *TriPoint* landmarks are numerous ($|\mathcal{M}^{(TriPoint)}| = 850$) and hence provide limited information about the position of the robot. $p(f_j|\mathbf{x})$ is modeled as

$$p(f_j|\mathbf{x}) \propto \beta + (1 - \beta)e^{-\frac{1}{2}((\frac{d}{\sigma_d})^2 + (\frac{a}{\sigma_a})^2)} \quad (1)$$

where d is the difference between the distances from sample position to map feature and robot to detected feature and a is the corresponding difference in angle. By comparing the data from the *TriPoint* detector with the map, a measure of the reliability can be estimated. This measure varies between different parts of the environment. Averaging at about 70% true positives, with about 50% in the worst case, $\beta_{TriPoint}$ is set to 0.5.

Planned Sampling The *TriPoints* have little value for planned sampling, first of all because a *TriPoint* feature will always match all map features of that type and second because each *TriPoint* has a large probable

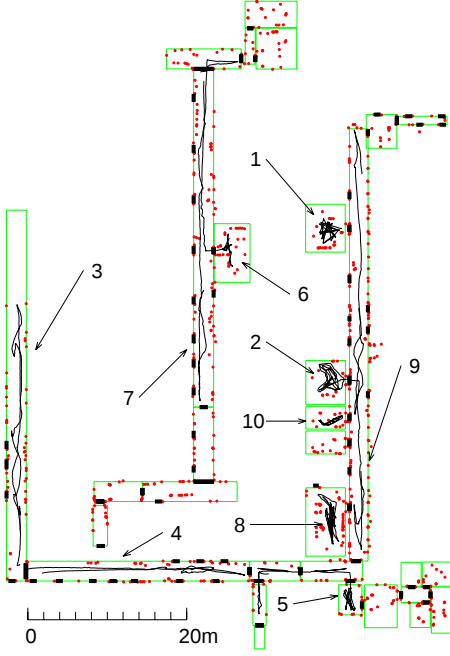


Figure 2: The map of features. The map consists of two floors connected by an elevator. Note that the rooms are modeled as rectangles. The doors are marked as thick lines and the *TriPoints* as small dots. The robot trajectories during the experiments are also shown.

area for the placement of planned samples (an annulus around the *TriPoint*).

A detected *TriPoint* is on average 1.5 m away from the robot, which gives

$$q_{TriPoint} \approx \frac{(1 - \beta_{TriPoint}) \cdot 0.5}{|\mathcal{M}(TriPoint)| \cdot (2\pi 1.5)} \approx 3 \cdot 10^{-5}.$$

4.2 Triangulation Point Pairs

By combining two triangulation points we obtain a more powerful feature, called *TriPair* henceforth. $p(f_j^{TriPair}|\mathbf{x})$ has two distinct peaks, resulting from intersecting the two annular PDFs from the *TriPoints*. The benefit of the *TriPair* feature is that it is better at producing planned samples, as each feature-map correspondence results in only two hypotheses, with small support areas. For ease of computation we have chosen to model the two intersections as Gaussians in the x-y- α plane, each centered at one of the intersection points of the two triangulation PDFs (compare Eq. 1). A *TriPair* is created if two *TriPoints* are further apart than $d_{min}^{TriPair} = 1.0$ m and closer to each

other than $d_{max}^{TriPair} = 10.0$ m. The map constructed using this criteria results in $|\mathcal{M}(TriPair)| = 16696$. Given that the *TriPoint* feature has $\beta_{TriPoint} = 0.5$, $\beta_{TriPair} = 1 - (1 - \beta_{TriPoint})^2 = 0.75$.

Correspondence between map and detected feature is deemed possible if the difference in distance is lower than a threshold which depends on the uncertainty in the position of the triangulation points.

Using *TriPairs* and *TriPoints* simultaneously in the update phase of the CONDENSATION algorithm clearly violates the assumption about independence between features. Thus, we used the *TriPoints* for updating the PDF and the *TriPairs* for creating planned samples.

Planned Sampling On average a *TriPair* is matched with 1300 map features which gives

$$q_{TriPair} \approx \frac{1 - \beta_{TriPair}}{1300} \approx 2 \cdot 10^{-4}$$

4.3 Lines

As seen in Figure 2, we assume that each room can be modeled as a rectangle. This is a very crude approximation, but one that has proven sufficient for pose tracking with good results in indoor office environments [11]. There are 31 rooms, giving $|\mathcal{M}(Line)| = 124$. Despite the fact that each room is modeled simply as a rectangle most of the *Lines* that are extracted from laser scans by a Hough transformation, correspond to lines that are in the map (approximately 50% which gives $\beta_{Line} \approx 0.5$).

The length of the line is used to determine possible correspondences between detected feature and the map features. The criteria for a match is

$$\text{length}(f_j^{Line}) < \text{length}(m_l^{Line}) + \text{length uncertainty}.$$

$p(f_j^{Line}|\mathbf{x})$ is high close to a line at the same distance from all matching map lines as the distance to the detected line (same as in [4]).

Planned sampling When the lines are used to produce planned samples, they result in samples being spread along lines parallel to all walls for which the above condition is fulfilled. On average 70 map features are matched, producing on average a 7 m long line along which the samples are spread.

$$q_{Line} \approx \frac{(1 - \beta_{Line}) \cdot 0.5}{|\mathcal{M}(Line)| \cdot 7} \approx 5 \cdot 10^{-4}.$$

4.4 Doors

The extraction of doors from laser data is based on a series of door models that capture different open door configurations. Only open doors can be detected. The doors that are in the map can be seen in Figure 2 as thick line segments on the walls. The map, $\mathcal{M}^{(Door)}$, does not capture the fact that some doors are very unlikely to be open. A more complete model of a door would involve the probability of the door being open.

We represent each door as two objects, one for each side of the door. This allows us to represent the fact that some doors can only be detected from one side (because the other side is inaccessible to the robot). Thus a *physical door* may correspond to one or two *Door* features in the map. The set of door map features has size $|\mathcal{M}^{(Door)}| = 100$ (for 71 physical doors).

A simplifying Gaussian assumption is made for $p(f_l^{Door}|\mathbf{x})$. The door is simply treated as a short line with direction from the right to the left door post. The same PDF is used as for lines, with the exception of a higher uncertainty in the position perpendicular to a line connecting the door post. This extra uncertainty corresponds to the wall thickness of approximately 0.5 m.

The *Door* detector produce up to 30% false positives in some environments. Therefore we set $\beta_{Door} = 0.3$.

Planned Sampling The door is by far the most powerful feature for planned sampling both because there are relatively few doors in the environment ($|\mathcal{M}^{(Door)}| = 100$), but also because there will be only one, rather distinct, peak in the PDF per map door feature (remember each physical door that can be passed through is represented by two map door features).

$$q_{Door} \approx \frac{1 - \beta_{Door}}{|\mathcal{M}^{(Door)}|} \approx 7 \cdot 10^{-3}$$

5 Experiments

To investigate how different combinations of feature types perform together or alone using CONDENSATION with planned sampling (see [8]), a series of runs with our Nomad200 robot equipped with sonar and laser sensors was conducted in different parts of an old hospital. The size of the sample set was 10000 and the number of planned samples was 200 per iteration.

Feature	$ \mathcal{M}^{(k)} $	q_k	Detection	Update
<i>TriPoints</i>	850	$3 \cdot 10^{-5}$	5 ms	48 μ s
<i>TriPairs</i>	16696	$2 \cdot 10^{-4}$	5 ms	9.1 μ s
<i>Lines</i>	124	$5 \cdot 10^{-4}$	20 ms	9.5 μ s
<i>Doors</i>	100	$7 \cdot 10^{-3}$	5 ms	16 μ s

Table 1: Size of the feature map, the relative quality measure and the computation time for detection and for updating the weight of one sample for the different features.

5.1 General Observations

The left part of the building, where run 3 was performed (see Figure 2), is almost completely without features, besides the two walls. Wall features constrain only two degrees of freedom and will not be sufficient for finding the pose of the robot. The same is true for the lower horizontal corridor next to the distance scale in Figure 2. This also applies to a lesser degree to all of the corridors in the map, in total over 200 m of corridor. A further property of corridors is that they tend to be less cluttered, offering fewer *TriPoints*. In our building in particular, the *TriPoints* in the corridor correspond largely to door posts and windows and are hence highly periodic.

5.2 Computation Time

In Table 1 the approximative computational time is given for extracting the different features as well as using them for updating the PDF. The detection time is for detecting all of a certain type of features from a data set, i.e. all lines in a laser scan or all triangulation points using the latest sonar readings. The update time is given for updating the weight of one sample. These values depend heavily on the representation being used. The *TriPairs* is a good example of this, having by far the largest map size, but still not using much time to update the PDF. However, we have not optimized the representation for *TriPoints* and so these are slower. Since the representation for the *TriPoint*-map can be improved in the same fashion as the *TriPair*-map, this difference must be considered small.

5.3 Accuracy

The accuracy that a feature will provide for the localization is to a large degree a question of how many dimensions it can establish conditions on. Let us consider a corridor wall again, this *Line* feature on its

Traj\feat	L	D	T	P	LD	TP	DP	DT	LT	LP	DTP	LTP	LDP	LDT	LDTP
1	10%	0%	10%	0%	10%	30%	0%	20%	100%	0%	20%	80%	10%	90%	80%
2	100%	0%	30%	0%	100%	100%	0%	90%	90%	100%	80%	100%	100%	100%	100%
3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	0%	0%	0%	0%	0%	0%	0%	0%	70%	40%	0%	70%	70%	80%	70%
5	80%	0%	10%	0%	70%	70%	0%	0%	100%	100%	0%	100%	100%	100%	100%
6	0%	0%	0%	0%	30%	0%	0%	0%	100%	40%	10%	100%	90%	100%	100%
7	60%	0%	10%	0%	30%	20%	0%	0%	100%	90%	20%	90%	70%	80%	100%
8	100%	0%	40%	20%	40%	100%	0%	0%	100%	100%	40%	100%	100%	100%	100%
9	20%	0%	0%	0%	90%	30%	0%	0%	90%	80%	60%	90%	100%	100%	100%
10	10%	0%	0%	0%	10%	0%	0%	20%	70%	30%	20%	100%	70%	80%	100%
av g.	38%	0%	10%	2%	38%	35%	0%	13%	82%	57%	25%	83%	71%	83%	85%

Table 2: Summary of running 10 experiments for each trajectory (from Figure 2) and feature combination, in total 1500 experiments. As no combination of features was enough to successfully localize along trajectory 3, 90% can be considered as the maximal achievable average. Character codes: L - *Line*, D - *Door*, T - *TriPoint* and P - *TriPair*.

own will only provide information about the perpendicular distance to that wall and the orientation of the robot, and it will do so with very high accuracy. Using *Line* features in environments where non-parallel *Lines* can be detected will provide excellent accuracy. *Doors* do not provide very good accuracy because of the thick walls in the old hospital building. *TriPoints* can provide excellent accuracy if the matching problem is solved, i.e. the correct map feature is matched to the detected feature, but this is difficult since the *TriPoint* map features have very little separation in some areas.

5.4 Convergence

For the convergence we check if the estimate converged to the correct pose and also how long time it took, using different combinations of features. Table 2 shows the results of running the CONDENSATION algorithm with planned sampling on different feature combinations (columns). The data was collected along the trajectories marked as 1 to 10 in Figure 2. The character combinations in the table indicate which features were used, L for *Line*, D for *Door*, T for *TriPoint* and P for *TriPair*. Trajectory 3 contained very few features (closed doors, little clutter, etc) and thus no combination of features provided enough information. The *Line* feature is clearly the best single feature. This is because *Lines* are easily detected and because there are few *Lines* in the map (in comparison to *TriPoints* and *TriPairs*). Unsurprisingly, the best results are achieved with a combination of features from different sensors.

5.5 Planned Sample Quality

The strength of the planned sampling modification of the CONDENSATION algorithm is in its ability to

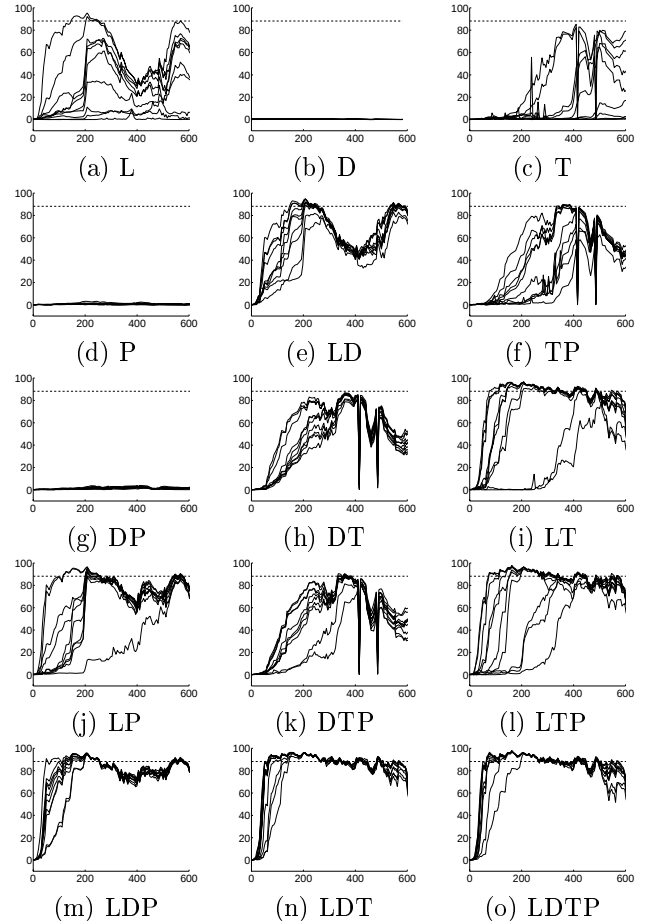


Figure 3: Each sub-figure shows the result of 10 attempts to perform global localization along trajectory 9 using different combinations of features (same notation as in Table 2). For each experiment the percentage of the total weight of the samples within 1 m of the true pose is shown on the vertical axis. The dashed lines mark the threshold used for a successful localization (see [8] for more details).

quickly gain sample support where $p(\mathbf{x}|\mathcal{F})$ is significant. Once support has been obtained at the true pose, the CONDENSATION algorithm handles the rest by itself. To get a high probability of placing a sample at the true pose, a feature should thus provide a peaked $p(\mathbf{x}|f_j^k)$. The result of high quality planned samples is that the algorithm starts to converge fast.

Judging from the relative quality measures of the different features (see Table 1), *Doors* are by far the best at producing planned samples. This is supported by Figures 3(a) and (e) where we can see that the combination of *Lines* and *Doors* converges to the true pose much faster than *Lines* alone.

5.6 Summary

From Figures 3(a)-(d) it is clear that one feature on its own does not provide enough information. In Figures 3(e)-(j) two features are used. All but the combination of *Doors* and *TriPairs* have improved the performance. The reason for the failure when combining *Doors* and *TriPairs* is that these features are not detected as frequently as the two other feature types. *Lines* and *TriPoints* have their strength in making the algorithm converge once support is given at the true pose. Thus, combining *Lines* or *TriPoints* with either *Doors* or *TriPairs* yields good results. The performance increases even more when using more features, with the best results when using all features.

As an example that the performance is improved by combining different features and sensors, we see from Table 2 that *Lines* on their own only give successful localization in 38% of the tests. The corresponding value for *TriPoints* is 10%, but when we combine them we get 82%.

6 Conclusions

In this paper we have presented four different features to be used with the CONDENSATION algorithm in combination with the planned sampling scheme presented in [8]. We conclude that all features require about the same amount of computation time. From the experimental comparison we conclude that *Lines* are the best single feature because they are frequently detected, but are not too common in the map. *TriPoints* are the next best feature because of ease of detection. There are few *Doors* and thus they are very good at providing the initial sample support at the true pose through planned sampling. *TriPairs* also have their strength mostly as a producer of planned samples.

For good results, features that are able to place samples close to the true pose are needed along with features that provide convergence. This might be one and the same feature, but in our case these qualities are given by different features. It is clear from the experimental results that it is very important to use different sensors to achieve robustness as a single sensor might not provide valuable information in the whole environment.

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. Borenstein, H. Everett, and L. Feng, *Navigating Mobile Robots: System and Techniques*. A K Peters, Ltd., 1996.
- [2] J. L. Crowley, "World modeling and position estimation for a mobile robot," in *IEEE Intl. Conf. on Robotics and Automation*, vol. 3, pp. 1574–1579, 1989.
- [3] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*. Boston: Kluwer Academic Publisher, 1992.
- [4] P. Jensfelt and S. Kristensen, "Active global localisation for a mobile robot using multiple hypothesis tracking," in *Workshop on Reasoning with Uncertainty in Robot Navigation (IJCAI'99)*, (Stockholm, Sweden), Aug. 1999.
- [5] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *Intl. Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
- [6] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *IEEE Intl. Conf. on Robotics and Automation*, pp. 1322–1328, May 1999.
- [7] W. Burgard, A. Derr, D. Fox, and A. Cremers, "Integrating global position estimation and position tracking for mobile robots: the dynamic markov localization approach," in *Proc. of the Intl. Symposium on Intelligent Robotic Systems*, 1998.
- [8] P. Jensfelt, O. Wijk, D. Austin, and M. Andersson, "Experiments on augmenting condensation for mobile robot localization," in *IEEE Intl. Conf. on Robotics and Automation*, 2000.
- [9] O. Wijk, P. Jensfelt, and H. Christensen, "Triangulation based fusion of ultrasonic sensor data," in *IEEE Intl. Conf. on Robotics and Automation*, vol. 4, (Leuven, Belgium), pp. 3419–24, May 1998.
- [10] O. Wijk and H. Christensen, "Triangulation based fusion of sonar data for robust robot pose tracking," *submitted to IEEE Transactions on Robotics and Automation*, 1999.
- [11] P. Jensfelt and H. Christensen, "Laser based pose tracking," in *IEEE Intl. Conf. on Robotics and Automation*, (Detroit, Michigan, USA), IEEE, May 1999.