2 RELATED WORK

In the literature, many researchers studied the observation model for probabilistic localization methods. The beam model was proposed by thrun et al (Thrun et al., 2005). This model is sometimes called raytracing or ray cast model because it relies on ray casting operations within an environmental model to calculate the expected beam lengths. Pfaff et al. (Pfaff et al., 2006) propose to change the measurement variance during localization. He suggested to use smooth likelihood functions during global localization and more peaked functions during position tracking. In his subsequent work, he suggested an improvement to the likelihood model by using location-dependent sensor model that explicitly takes the approximation error from the sample-based representation into account (Pfaff et al., 2007).

In his later work, he uses Gaussian mixtures model GMM to model the likelihood function for single range measurements (Pfaff et al., 2008b; Pfaff et al., 2008a). It is place dependent likelihood. Plagemann et al. (Plagemann et al., 2007) use the Gaussian process instead of GMM. The advantage of the Gaussian process is that does not tend to get stuck in local minima as GMMs do. Both systems (Plagemann et al., 2007), (Pfaff et al., 2008a) still rely on an underlying gaussian distribution of the state space and are sensitive to discontinuities in the map (Plagemann et al., 2007), (Pfaff et al., 2008a). (Olufs and Vincze, 2009) present an area-based observation model. The model is based on the idea of tracking the ground area inside the free space (not occupied cells) of a known map. To reduce the effects of outliers (Yilmaz et al., 2010) calculates the total sensor probability through replacing the process of multipling individual sensor probabilities by individual probabilities geometric mean after repealing some extreme measurements.

In this research an improvement in calculating the likelihood function is proposed through neglecting the measurements that suspected to invalidate the independent noise assumption.

Before introducing the MCL for robot localization, the entropy filter and distance filter where proposed for dynamic environments with Markov Localization (Burgard and Thrun, 1999). The distance filter has been specifically designed for laser range finders. It rejects the measurements that have high probability to be shorter than expected. The disadvantage of this algorithm is that the calculation of the probability for each measurement requires computing all the ray casing for all proposal samples.

3 MONTE CARLO LOCALIZATION

Localization is the process of estimating the robot pose (position and orientation) in a given environment map. Assume at time instant t, the robot pose x_t consists of the robot x, y coordinates and the orientation angle θ

$$x_t = \{x, y, \theta\} \tag{1}$$

The environment is stored in occupancy grid map m. The measurement vector is denoted by z. It consists of K beams. The number over z represents the beam index not the power.

$$z_t = \left\{ z_t^1, z_t^2, ..., z_t^k \right\} \tag{2}$$

To estimate the pose x_t of the robot in its environment Monte Carlo Localization (MCL) is used. It is a probabilistic localization, which follows the recursive Bayesian filtering scheme (Dellaert et al., 1999).The recursive Bayes filter calculates the belief $bel(x_t)$ at time t from the belief $bel(x_{t-1})$ at time $t-1$ using the following equations (Thrun et al., 2005):

$$\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}) bel(x_{t-1}) dx_{t-1} \tag{3}$$

and

$$bel(x_t) = \eta p(z_t|x_t, m) \overline{bel}(x_t) \tag{4}$$

Here η is a normalization constant ensuring that $bel(x_t)$ sums up to one over all x_t.

The term $p(x_t|u_t, x_{t-1})$ describes the probability that the robot is at position x_t given it executed the movement u_t at position x_{t-1}. This is called also the probabilistic motion model. Furthermore, the quantity $p(z_t|x_t, m)$ denotes the probability of making observation z_t given the robots current location is x_t. This is called the observation model which is the important part in this research.

The Bayes filter possesses two essential steps. Eq.3 is called the *prediction* since the belief $\overline{bel}(x_t)$ is calculated from integrating the product of the previous belief and the probability that the control u_t make a transition from x_t to x_{t-1}.

Eq.4 is called the *measurementupdate* or *correction*. The belief $\overline{bel}(x_t)$ is multiplied by the probability that the measurement z_t may have been observed.

A sample-based (or particle-based) implementation of this filtering scheme is the Monte Carlo localization (Dellaert et al., 1999; Fox et al., 1999; Thrun et al., 2001). In Monte-Carlo localization, which is a variant of particle filtering, the belief $bel(x_t)$ is approximated by the set of particles χ_t. The update of the belief is realized by the following two alternating steps(Thrun et al., 2005):

1. In the *prediction* step, for each sample draw a new sample according to the weight of the sample and according to the model $p(x_t|u_t,x_{t-1})$ of the robots dynamics given the action u_t executed since the previous update.

2. In the *correction* step, the new observation z_t is integrated into the sample set. This is done by re-sampling. Each sample is weighted according to the likelihood $p(z_t|x_t,m)$ of sensing z_t given by its position x_t. More details about the re-sampling process and particle filters in (Dellaert et al., 1999; Fox et al., 1999; Thrun et al., 2001).

4 BEAM OBSERVATION MODEL

Several sensor models are based on the basic concept of the well known *beam model* that is used in traditional MCL methods such as Adaptive Likelihood Model ALM (Pfaff et al., 2006), Gaussian Beam Process model GBPM (Plagemann et al., 2007), Gaussian mixture models GMM (Pfaff et al., 2008a). The beam model considers the sensor readings as independent measurement vector z and represents it by a one dimensional parametric distribution function depending on the expected distance in the respective beam direction. This model is sometimes called raytracing or ray-cast model because it relies on ray casting operations within an environmental model e.g., an occupancy grid map, to calculate the expected beam lengths. The measured range value will be denoted by z_t^k while the range vale calculated by ray-casing will be denoted by z_t^{k*}. The model calculates the likelihood of each individual beam to represent its one-dimensional distribution by a parametric function depending on the expected range measurement. It is a

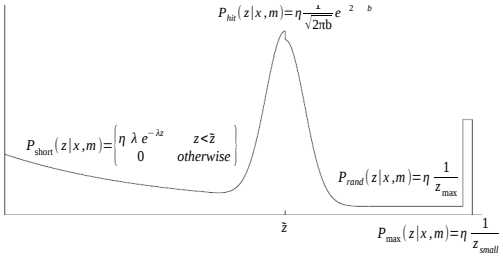

Figure 1: Single beam distribution.

mixture of four distributions. Gaussian distribution *Phit* for small measurement noise, exponential distribution *Pshort* for unexpected objects, uniform distribution *Pmax* due to failures to detect objects and uniform distribution *Prand* for unexplained noise.

$$Phit(z_t^k|x_t,m) = \begin{cases} \eta N(z_t^k;z_t^{k*},\sigma_{hit}^2) & \text{if } 0 \le z_t^k \le z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The term $N(z_t^k;z_t^{k*},\sigma_{hit}^2)$ denotes the univariate normal distribution with mean z_t^{k*} and standard deviation σ_{hit}^2.

$$Pshort(z_t^k|x_t,m) = \begin{cases} \eta\lambda_{short}e^{-\lambda_{short}z_t^k} & \text{if } 0 \le z_t^k \le z_t^{k*} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$Pmax(z_t^k|x_t,m) = \begin{cases} 1 & \text{if } z = z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$Prand(z_t^k|x_t,m) = \begin{cases} \frac{1}{z_{max}} & \text{if } 0 \le z_t^k < z_{max} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

The η is a normalization factor that is calculated to make the distribution integrates to one in its interval. The four distributions are mixed by a weighted average defined by the parameters z_{hit}, z_{short}, z_{max} and z_{rand} such that $z_{hit}+z_{short}+z_{max}+z_{rand}=1$

$$P(z_t^k|x_t,m) = z_{hit}P_{hit} + z_{short}P_{short} + z_{max}P_{max} + z_{rand}P_{rand} \quad (9)$$

The parameters z_{hit}, z_{short}, z_{max}, z_{rand},σ_{hit} and λ_{short} can be set by hand or calculated from a stored data using an estimator. More details about these distributions, their expressions and parameters in the book (Thrun et al., 2005).
The total measurement probability is the multiplication of all individual beams probabilities

$$P(z_t|x_t,m) = P(z_t^1|x_t,m)\cdot P(z_t^2|x_t,m)\cdot\ldots\cdot P(z_t^K|x_t,m) \quad (10)$$

$$P(z_t|x_t,m) = \prod_{k=1}^{K} P(z_t^k|x_t,m) \quad (11)$$

This is based on the independence assumption between the noise in each individual measurement beam. Unfortunately, this is true only in ideal cases. Dependency typically exist due to: people, unmodeled objects and posterior approximations that corrupt measurements of several adjacent sensors (Thrun et al., 2005). This kind of noise will result in peaks in the likelihood measurement function. These peaks leads to incorrect particles distribution in the MCL. This is because the particles are redistributed proportional to the importance weights, and these weights are calculated from the likelihood function. Therefore, the peaked likelihood will result in concentrating the particles on the same state during the re-sampling

process. Then more particles are required to restore the distribution.
The divergence between Eq.3 and Eq.4 determine the convergence speed of the algorithm (Thrun et al., 2001). This difference is accounted by the observation model, however, if the $p(z_t|x_t,m)$ quite narrow (has peaks) then the MCL converges slowly.
Mathematically, peaks in this likelihood function are generated due to the Normal distribution used in the observation model see Fig.1and Eq.5, therfore, a small difference in pose between the proposal and actual x_t will results in large differences in the likelihood function. In practice, peaks could be generated for different reasons: high accuracy sensors, highly cluttered environment, incomplete or imperfect map and moving people or dynamic obstacles that invalidate the noise independence assumption as stated above.

One solution to over come the problem of independent noise assumption is to calculate the conditional probability between the beams as shown in Eq.12.

$$\begin{aligned} &P(z_t|x_t,m) = P(z_t^1|z_t^2,z_t^3,...,z_t^k,x_t,m) \\ &\cdot P(z_t^2|z_t^3,z_t^4,...,z_t^k,x_t,m) \ \ ... \cdot P(z_t^{K-2}|z_t^{K-1},z_t^K,x_t,m) \\ &\cdot P(z_t^{K-1}|z_t^K,x_t,m) \cdot P(z_t^K|x_t,m) \end{aligned} \quad (12)$$

These conditional probabilities could be difficult to compute. Another solution to this problem that is easier to implement is to remove those beams that invalidate the independent noise assumption through prefiltering or sub-sampling the scan beams before doing the MCL.

5 PROPOSED BEAM REDUCTION SCHEME

In this research, an adaptive sub-sampling of the measurements is proposed in the likelihood function. The sampling is based on the complete scan analysis. It is adaptive, because the sampling may differ from scan to another depending on the measurements itself. The specified measurement is accepted or not based on the relative distance to other points in the 2D point cloud. Instead of using all of the 'K' beams in Eq. 11, adaptively reduce the number of beams in each scan according to the distance between returned measurements from adjacent scan beams (euclidean distance in this case)

$$\begin{aligned} &P(z_t|x_t,m) = \prod_{k=1}^{K} \overline{P}(z_t|x_t,m) \\ &\overline{P}(z_t|x_t,m) = \begin{cases} 1 & \text{if } |z_t^k - z_t^{k-1}| \leq \ \delta \\ P(z_t|x_t,m) & \text{if } |z_t^k - z_t^{k-1}| > \ \delta \end{cases} \end{aligned} \quad (13)$$

The δ is the distance threshold value that specifies the smallest distance to next accepted point in the 2D space. Based on the assumption that the very close points in the 2D space are most likely to be reflected from the same obstacle. Beams reflected from the same obstacle may invalidate the independent noise assumption in cases like the obstacle is not modelled in the map (incomplete map) or it is a walking person. On the other hand, reducing the number of beams will reduce the computations required in the localization process. In fact, the raycasting process is a very computation expensive process. Therefore, reducing the number of beams will reduce the computations effectively. Of course, some useful beams with independent noise will also be removed by the algorithm. This will slightly affect the localization process accuracy. However, there will always be other beams close to the removed ones.

6 SIMULATION RESULTS

The effect of changing δ values on the observation model likelihood function peaks was analyzed and demonstrated using real data recorded from SICK LMS laser scanner mounted over Pioneer 3AT robot. The robot position was fixed during this test. The robot situated inside normal office environment. The data recorded with and without moving person in the field of view see Fig.2 . To analyze the peaks, the

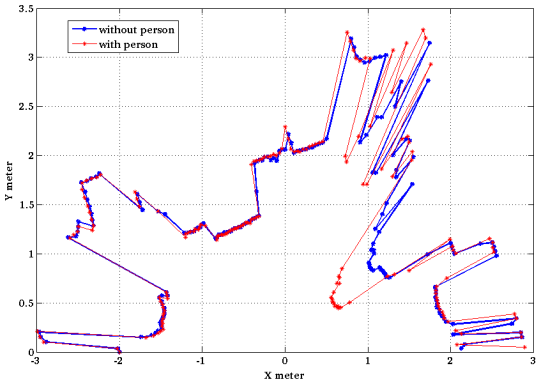

Figure 2: Laser scan in typical office environment.

likelihood function has been calculated between each scan and the whole data set. Fig.3 shows the ratio between the first to the second peaks in the likelihood function for all the recorded scans. Again, the robot position is fixed so ideally there should be no dominant peaks since the pose is not changing. The only change is in the moving person position and the measurement noise. It is clear that larger values of δ result

in less dominant peaks. The segments with no peaks in the figure represents the scans recorded when the moving person is out of the scanner field of view.

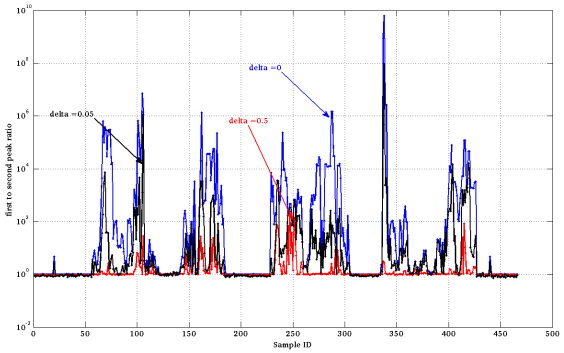

Figure 3: First to second peak ratio in likelihood function for δ = 0,0.0.5 and 0.5; blue black red respectively .

The described reduction scheme has been verified using Robot Operating System (ROS) fuerte release (Willow Garage, 2012) and stage simulator (Vaughan, 2008). The standard ROS monte-carlo localization algorithm (amcl) has been used with modifications in the observation model. The global localization has been used in all simulations. The initial robot position is assumed to be unknown. The robot moves in two paths, the first path is 8 shape path while the second one is c shaped path. The robot is assumed to operate inside a small room with only two fixed rectangular obstacles and 8 umodeled obstacles see Fig. 4.

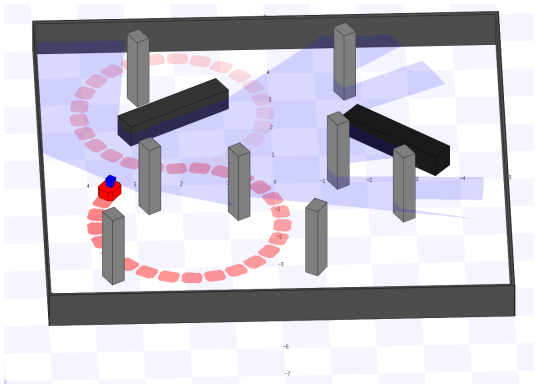

Figure 4: simple room with 2 modeled obstacles (black) and 8 extra unmodeled obstacles (gray).

A map of this environment with only two fixed obstacles was build using the slam gmapping node in ROS with resolution 0.050 m/pix. The robot translation speed was set to 1m/s. The number of particles was fixed to 1200 particles. The simulations divided into three groups:

6.1 No Extra Obstacles

Fig 5 shows the results obtained from the 8 shape path and no extra obstacles case. It is clear from the blue line that for δ values less than 1, the translation error is reduced as delta is increased. A δ value around 0.7 is giving the best combination between translation error and computation. All the measurements are normalized with the values obtained from the original monte-carlo localization algorithm (i.e. the case that δ = 0). The blue line is the normalized translation error, the red line is the normalized rotational error and the yellow line is the normalized computation reduction.

Fig 6 shows the same case but with C shape path.

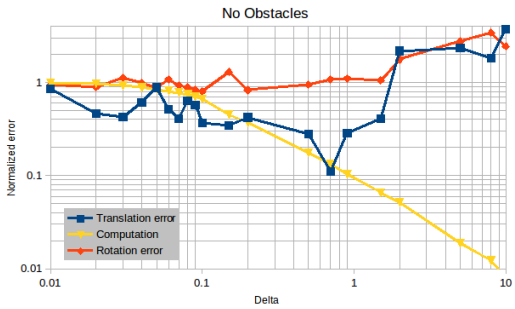

Figure 5: Normalized translation error, rotation error and computations versus δ value with no extra obstacles.

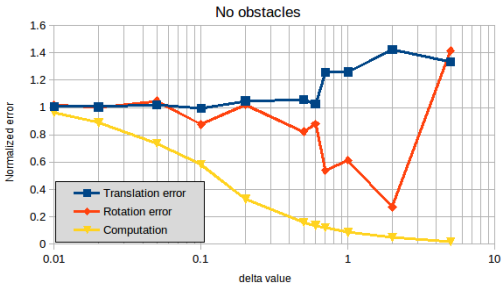

Figure 6: Normalized translation error, rotation error and computations versus δ value with no extra obstacles .

6.2 Extra Stationary Obstacles

Fig 7 shows the results obtained from the 8 shape path with extra obstacles case. Fig 8 shows the same case but with C shape path.

6.3 Extra Moving Obstacles

Fig 9 shows the results obtained from the 8 shape path with extra moving obstacles case. Fig 10 shows the same case but with C shape path.

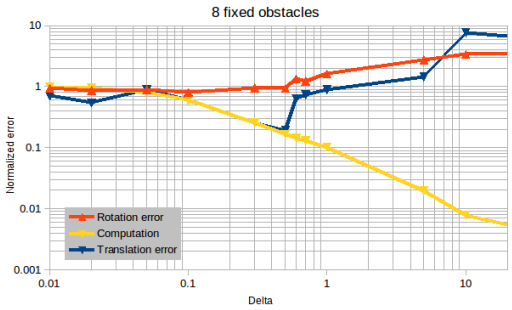

Figure 7: Normalized translation error, rotation error and computations versus δ value with 8 extra obstacles.

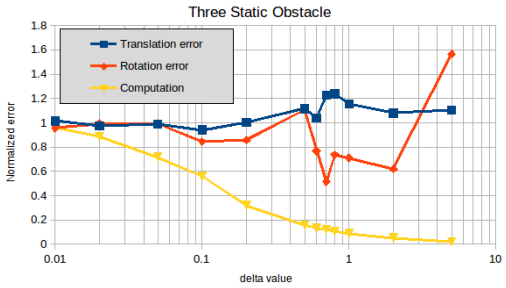

Figure 8: Normalized translation error, rotation error and computations versus δ value with 8 extra obstacles.

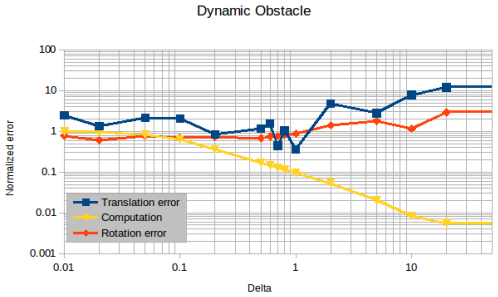

Figure 9: Normalized translation error, rotation error and computations versus δ value with moving obstacles.

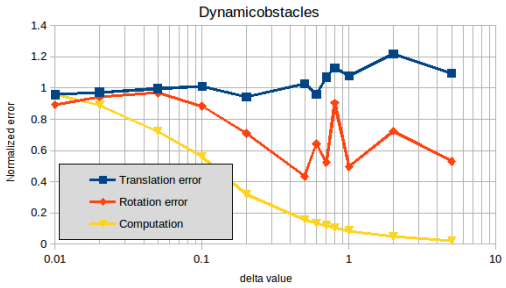

Figure 10: Normalized translation error, rotation error and computations versus δ value with moving obstacles.

It is clear from all the figures above that applying this simple reduction scheme results in a noticeable improvement either in the translation error or in the rotational error beside approximately 50% reduction in computation complexity.

6.4 Trajectory Error

To investigate the performance of the proposed algorithm the trajectory error for three different cases is shown below. The trajectory error is computed as the difference between the position estimation and the ground truth for each time sample. In each case a comparison with the standard MCL and MCL with distance filter proposed by (Burgard and Thrun, 1999) is presented.

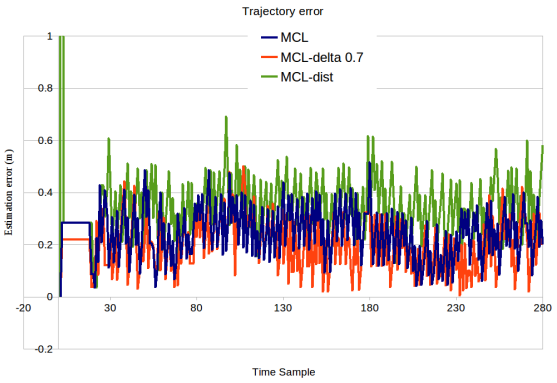

Figure 11: Trajectory error comparison for no obstacles case.

Fig. 11 shows the case where no unmodeled obstacles is presented. The estimation error for the proposed algorithm is clearly lower than the others. Also it is clear that the MCL with distance filter has higher estimation error.

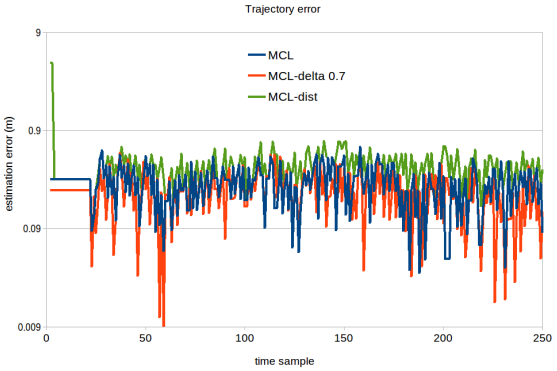

Figure 12: Trajectory error comparison for moving obstacles case.

Fig. 12 shows the case where dynamic obstacles are presented in the environment. Again the estimation error for the proposed algorithm is clearly lower than

the others and the MCL with distance filter has higher estimation error.

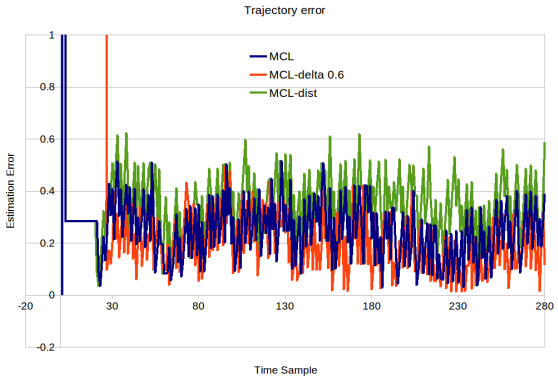

Figure 13: Trajectory error comparison for static obstacles case.

Fig. 13 shows the case where static but unmodeled obstacles are presented in the environment. In this case also the estimation error for the proposed algorithm is clearly lower than the others and the MCL with distance filter has higher estimation error.

6.5 Computation Complexity

The Table below shows the computation complexity of the proposed technique compared with traditional MCL and MCL with distance filter. It is very clear that there is a very large reduction in computations. The number represents how frequent the ray casting function is called. It is normalized to standard MCL value. The δ value was set to 0.7 in this table.

Table 1: Computation Complexity reduction table.

Standard MCL	MCL with Distance filter	MCL with proposed scheme
1	1.994	0.12832

7 CONCLUSION AND FUTURE WORK

In this research a modification to observation model that is used in Monte Carlo localization has been proposed. This modification reduces the peaks generated in the observation likelihood function that is limiting the performance of the localization process. Specifically the peaks generated due to the invalidation of the independent noise assumption between different measurement beams.
The proposed scheme has been verified using Stage simulator and ROS Robot operating system. the results shows improvement in both location estimation error and the computations required for localization. As future work, it is possible to find a reliable method to calculate or estimate the conditional probabilities in Eq.12 instead of neglecting some measurements. Also it is important to find solutions to other causes of the peaks in the observation model. This will definitely improves the localization farther.

REFERENCES

Burgard, W. and Thrun, S. (1999). Markov localization for mobile robots in dynamic environments dieter fox dfox@ cs. cmu. edu computer science department and robotics institute carnegie mellon university. *Journal of Artificial Intelligence Research*, 11:391–427.

Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE.

Fox, D., Burgard, W., Dellaert, F., and Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999:343–349.

Olufs, S. and Vincze, M. (2009). An efficient area-based observation model for monte-carlo robot localization. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 13–20. IEEE.

Pfaff, P., Burgard, W., and Fox, D. (2006). Robust monte-carlo localization using adaptive likelihood models. In *European robotics symposium 2006*, pages 181–194. Springer.

Pfaff, P., Plagemann, C., and Burgard, W. (2007). Improved likelihood models for probabilistic localization based on range scans. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2192–2197. IEEE.

Pfaff, P., Plagemann, C., and Burgard, W. (2008a). Gaussian mixture models for probabilistic localization. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 467–472. IEEE.

Pfaff, P., Stachniss, C., Plagemann, C., and Burgard, W. (2008b). Efficiently learning high-dimensional observation models for monte-carlo localization using gaussian mixtures. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3539–3544. IEEE.

Plagemann, C., Kersting, K., Pfaff, P., and Burgard, W. (2007). Gaussian beam processes: A nonparametric bayesian measurement model for range finders. In *Robotics: Science and Systems*.

Rowekamper, J., Sprunk, C., Tipaldi, G. D., Stachniss, C., Pfaff, P., and Burgard, W. (2012). On the position accuracy of mobile robot localization based on particle filters combined with scan matching. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3158–3164. IEEE.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press.

Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1):99–141.

Vaughan, R. (2008). Massively multi-robot simulation in stage.

Willow Garage, S. A. I. L. (2012). The robot operating system.

Yilmaz, S., Kayir, H. E., Kaleci, B., and Parlaktuna, O. (2010). A new sensor model for particle-filter based localization in the partially unknown environments. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 428–434. IEEE.