The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

# An Efficient Area-based Observation Model for Monte-Carlo Robot Localization

Sven Olufs and Markus Vincze

*Abstract*— **The problem of mobile robot self-localization is considered as solved since Thrun's et. al [1] pioneering work using monte-carlo filters for robot Localization (MCL). However, MCL is robust and precise under constraints like completely known environments and the sensor data must contain enough "true data" as contained in the map. In fact these conditions cannot always be guaranteed, which may results in a poor accuracy of the localization.**
**In this paper we present a area-based observation model that is applied to MCL self-localization. The model is based on the idea of tracking the ground area inside the "free space" (not occupied cells) of a known map. Experimental data shows that the proposed model improves the robustness and accuracy of laser and stereo vision sensors under certain conditions like incomplete map, limited FOV and limited range of sensing. We also present an efficient approximation of our sensor model based on integral images.**

## I. INTRODUCTION

Robust self-localization is the base for many behavior-based systems. It has to cope with uncertainty due to imprecise sensors, incomplete knowledge of the environment and limited range of sensing. Very popular approaches for localization are the dynamic state space models using bayesian filter variants like monte-carlo filters (MCL) [1] or extended Kalman filters (EKF) [2]. They address the problem of estimating the state $x$ of a *dynamic system* from measurements and sensor readings. Control theory describes a *dynamic system* [3] as an interaction model between a *controller* with its environment. Both entities interact with each other through *signals* $y$ (e.g., Laser range data) and *actions* $u$ (e.g., Odometry). A *motion model* is used to integrate the *actions* $u$ into the current pose/state while the *observation model* integrates the observations. In this paper we will focus on the *observation model* for MCL filters with range sensors.
The observation model, i.e., $p(y|x,M)$ is the heart of many self-localization system. It is a likelihood function that specifies how to compute the likelihood of an observation $y$ given a believed robot pose $x$ in an a-priori known map $M$. The function is often a trade-off between accuracy and robustness. A restrictive model may produce good accuracy with monte-carlo filters, but at the cost of robustness and vice versa. Other filters like EKF may use a restrictive observation model to avoid mis-associations which can have side effects on the filters itself. The literature proposes several observation models for specific sensors and applications, e.g., models able to cope with sonar crosstalk [4] or omni-directional vision [5]. Many observation models are based on the basic concept of the well known *beam model* [1], [3] that is also used in traditional MCL methods. The beam model considers the sensor readings as independent measurement vector $y$ (e.g., distance and angle) and represents it by a one dimensional parametric distribution function depending on the expected distance in the respective beam direction.

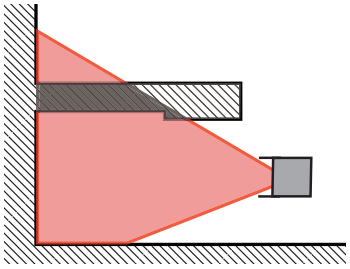


Fig. 1. Basic Concept of the proposed Model. The red areas are used in the individual models for calculating the likelihood $p(y|x,M)$.

The well known MCL method has been proven a robust and precise method [3], [1], [6], but only under certain constraints: The map must contain the complete environment and the sensor data must contain enough "true data" that is known from the map (See section VI). In fact not all constraints can always be fulfilled due to, e.g., technical restrictions of the sensor or changing environment. A map that does not contain all features as well as limited sensing (Field of view or range) often lead to unsatisfactory accuracy.

In this paper, we propose a new area-based observation model that tracks the ground area inside a the "free space" (that is, the not occupied cells) of a known map by using (extracted) visibility information according to the believed pose to the map. In practical experiments carried out with data obtained by a real robot we demonstrate that our model improves the accuracy and robustness of standard MCL methods when dealing with incomplete maps and limited range of sensing. We use laser and raw stereo vision data as sensors in order to demonstrate the robustness of our model. We also propose an efficient real-time approximation of the observation model that is based on *integral images* and image decomposition.

The research leading to these results has received funding from the European Community's Sixth Framework Programme (FP6/2003-2006) under grant agreement no FP6-2006-IST-6-045350 (robotshome)

Sven Olufs and Markus Vincze are with the Vienna University of Technology, Automation and Control Institute, Gusshausstrasse 30 / E376, A-1040 Vienna, Austria

The paper is organized as follows: After discussing the state of the art, we describe in section III the motivation for using a volumetric model. This is followed by a description of the volumetric observation model for self-localization. In section V an efficient approximation of our approach is given. Experimental results are presented in Section VI. Finally, we discuss the results and the approach in Section VII.

## II. RELATED WORK

The literature proposes various techniques for computing the likelihood of the observation model. They either directly approximate the physical characteristics of the sensor or use models that try to provide smooth likelihood models to increase the robustness of the localization process. A well known variant of the first type of model is the *beam model*, by Thrun et al [1]. 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. 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. Bennewitz et al. [7] extend the standard raytracing technique by utilizing reflection properties of the sensor and surfaces by incrementally learning the environment. Pfaff et al. [8] propose an approach for multi-modal likelihood Models by using Gaussian Mixture Models to learn the observation Model. The GMM does two things: It smoothes the observation model for better robustness to outliers and compresses the state space due to its parametric representation. Plagemann et al. [9] 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 [8], [9] still rely on an underlying gaussian distribution of the state space. In fact both models are sensitive to discontinuities in the map i.e. when the environment is cluttered or the map is not precise [10], [8], [9]

A different approach to observation models is the so called *likelihood field* technique [10], [11] or *beam end point model*. It provides smooth and multi-modal likelihood functions to better deal with clutter in the environment. These kind of model ignores the physical constraints on purpose to improve the robustness on cost of accuracy [9]. The likelihood of a single sensor reading depends on the distance of the corresponding endpoint to the closest wall in the map. These models suffer from two drawbacks. First, they do not take visibility constraints into account. Second, they provide no direct mechanism to deal with maximum-range readings, which is why these readings are typically ignored.

## III. MOTIVATION

### A. Incomplete Map

Depending on the type of robot application, a map for robot localization can be incomplete, i.e., it does not contain all objects like chairs, side table or drawer. This can happen accidentally, because objects were missing during map building or on purpose due to practical reasons, i.e., movable objects might confuse localization.

### B. Short Sensing Range of the Sensor

Recent developments in range sensors have produced a new series of laser scanners, e.g., the Hokuyo URG-04LX. This kind of sensors is quite popular in robotics due to their small size, weight, power consumption and price. The relatively short sensing range, e.g., max 4m for the URG-04LX is a clear constraint compared to the well known SICK LMS 200 with up to 80m sensing range (30m typically).

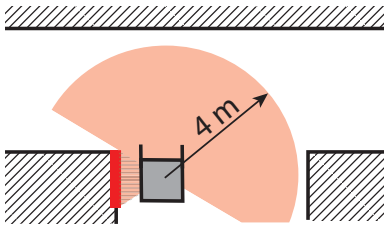



Fig. 2. Hokuyo URG-04LX on the robot in a spacious environment: The circular area shows the FOV and sensing range of the sensor. The thick red line represents the only wall segment that is detected, the rest is out of range.

In Figure 2 the localization accuracy is unsatisfactory, because only the wall on the left is in range of the sensor. Traditional feature-based approaches always expect a distance and angle to the observed feature, e.g., the wall or a wall segment. The sensor readings that report *out of range* do not report a true observed distance. Hence they cannot be used. In the Example, Figure 2, about 75% of the sensor data reports *out of range* and is not used in a feature-based method.

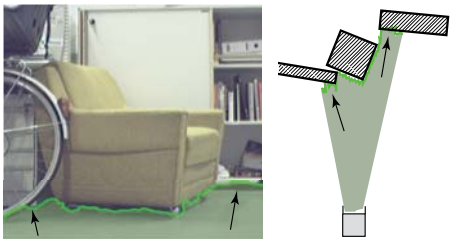
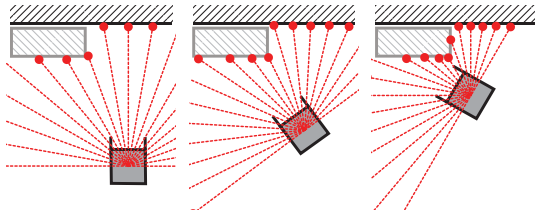


Fig. 3. Stereo vision as range sensor: The left picture shows the input camera image. The green area represents the extracted ground with stereo vision.

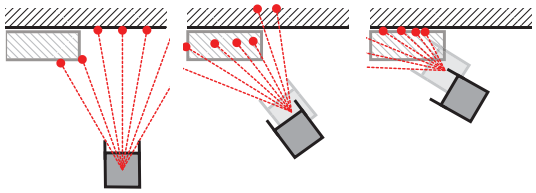
The short range sensing problem is not bounded to low cost laser range sensors. This problem also appears with vision based range sensors for feature extraction, e.g., using stereo vision to detected wall segments. Due to the nature of stereo cameras, i.e., motion blur, false matching, poor resolution it is not always possible to detect a wall segment. Let us consider Figure 3: The detected wall segments with a high confidence have been marked with arrows. The little wall segment information is due to the clutter in the scene and motion blur. It is possible to extract at least a fragment of the ground plane.

### C. Narrow FOV of the sensor

The use of camera-based solutions for navigation and localization has become quite popular in the last decades. Compared to typical laser range sensors cameras provide a rich set of information [12], [13], e.g., texture. However, these sensors typically have a narrow field of view, e.g., stereo cameras up to $110^o$ (typically $50^o$) or Time-of-Flight cameras like MESA Imaging's SwissRanger $\approx 45^o$ FOV. A narrow FOV can lead to unwanted side effects, if the known map of the environments is incomplete: Figure 4 demonstrates such a behavior with a wide and narrow FOV. The narrow-FOV sensor starts to consider the unmapped box as the wall if no true wall segments are observed. The wide-FOV sensor is more robust with respect to scan points produced by the box, because the scan contains enough data from the true known wall (See Figure 3)



(a) Field of View $180^o$



(b) Field of View $60^o$

Fig. 4. Unwanted behavior due to narrow FOV and incomplete map: The wall in front of the robot is mapped, the box on the left is not. The robot moves through the corridor and estimates the pose with a laser range scanner. In both subfigures the estimated pose is shown, the true position is hinted by a transparent robot. One can see that the robot with limited FOV starts to consider the box as the wall if less than 50% of the true wall is observed. The robot with the wide FOV is able to track the true wall as enough sensor readings of the true wall are received.

## IV. PROPOSED OBSERVATION MODEL

The model is inspired by the work of Thrun [14] on robot mapping using a *ground space* model for local maps. The *ground space model* is used for alignment, but not for localization itself since the ground cannot be transformed into e.g. laser-like scan. Such a scan contains measured distances to objects while the *ground space* model contains only the free space that does not necessarily corresponds to the distance to an certain object, for instance with distances "out of sensor range". The main idea of our observation model is as follows: We detect the *ground space* area and keep the detected space inside the non-occupied area of the a-priori known map, see Figure I. In the fashion of bayesian filters and Markov Chains we track and refine the pose over time. It is assumed that the ground can be detected rather than detecting a wall. We model the *ground space* in contrast to standard localization approaches in an opportunistic way: We define an observation model that calculates the difference of the observed ground space and the free space according to the map at the believed pose. This approach implicitly deals with unmapped regions as well. For instance, the result is zero, if the believed ground is actually in a non-occupied ($\approx$ free) part of the map. The error grows if the believed area is actually not in the *free space* of the map.
As result we obtain possible poses that are not violated (i.e., not wall/unexplored areas are intersected) by the observed area. In combination with a state estimation algorithm and motion model we are able to track the pose (see section IV-B).

### A. Observation Model

We propose the *Observation Model* $p(y_t|x_t)$ in bayesian fashion [3]: Let $x$ be the believed state in an n-dimensional state space and $y$ the observation from a range finder sensor, e.g., stereo vision, sonar. The observation model reflects the probability of measuring $y_t$ if we assume that the state of the robot is $x_t$. Let $M$ be the a-priori known occupancy grid map that is used for localization. It contains information about free space (absence of objects), solid objects (e.g., walls) and unexplored space. Let $A$ be the *ground space* that has been observed by the sensor measurements $y$.

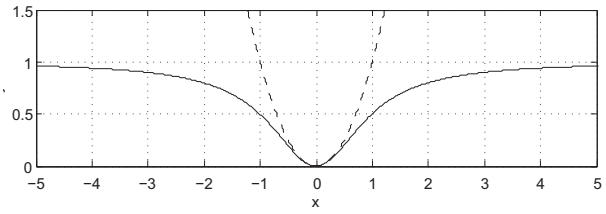



Fig. 5. Comparison of the squared error function (dashed line) and the more robust M-estimator $e \rightarrow 1 - \frac{c^2}{c^2+e^2}$ (solid line)

First, the map $M$ is converted into a cost matrix $K$ in the spirit of the *likelihood fields* [10], [15] technique. The costs of *unexplored* and *solid object* areas are calculated first. The Euclidean distance of each "wall" cell to its nearest *free cell* of the map $M$ is used as metric. Let $e$ denote the distance of a point from an *unexplored* or *solid object* area in $M$. $e$ is applied to a weight function: The squared error function $e \rightarrow \frac{1}{2}e^2$, which is standard for many applications, is not appropriate for the given task since it is not robust with respect to outliers [16]. Due to noise and imprecision of the sensors, i.e., crosstalk with sonars or false correspondence matching with stereo cameras, there is a substantial amount of erroneously detected ground space that would distort the estimate. Instead, we use the error function $e \rightarrow 1 - \frac{c^2}{c^2+e^2}$ with parameter $c = 100$ $(1m)$, see fig. 5. This error function is very similar to the squared error function for errors $e \leq c$ and is bounded above by a constant for larger errors, thus the influence of outliers onto the estimate is bounded.

Let $k_{1..n} \in K|x_t$ be elements that represent the *ground space A* with $n$ elements in the cost matrix $K$. We use the following model to calculate the likelihood of the sensor model

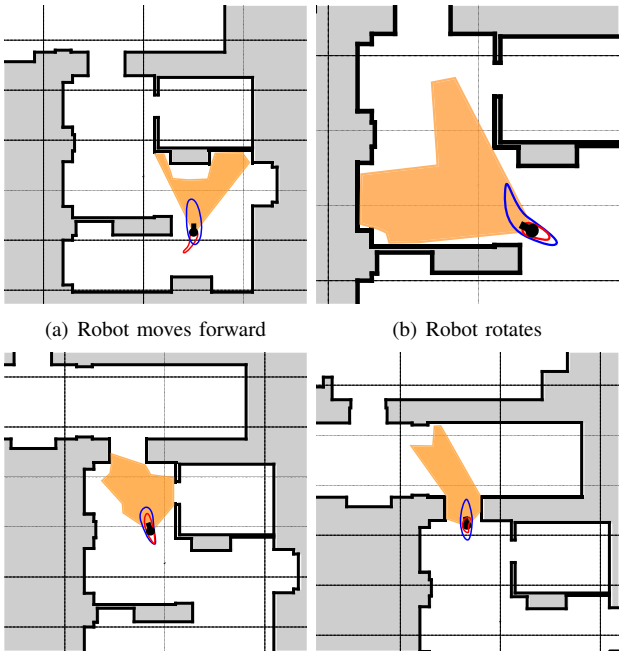
$$p(y_t|x_t) = \eta \sum_{i=0}^{n} [1.0 - B(k_i|x_t)] \quad (1)$$

with $\eta$ as normalizing constant. Here it is also assumed that all observations are independent from each other. $B(k_i|x_t)$ does a lookup in the cost matrix $K$ and returns the likelihood of the observation model $p(y_t|x_t)$. In contrast to the approach in literature [6] we are summing up the probabilities instead of multiplying them. This results in a lower sensitivity of the probability to (few) false-positive observations.

### B. Applying the model to Localization

In theory the observation model can be applied to all kinds of Bayesian localization methods, but we recommend the usage of a multi-hypothesis system, e.g., MCL. We use the standard MCL motion model using non-/holonomic constrains [3]. With our observation model MCL is able to perform an initial global localization, but more iterations are needed to find the initial pose (like when using sonar sensors compared to a laser scanner). It is important that the robot has to move or rotate in order to refine the pose and eliminate the false hypotheses (i.e., ground space that intersects the walls). The usage of gaussian noise on the particles is the same as in motion models used for MCL.

The translation part of the pose is updated with ground space observations if it contains a true distance to (a part of) a wall.



(a) Robot moves forward (b) Robot rotates

(c) Robot moves forward (d) Robot rotates

Fig. 6. MCL with the proposed model in practice with the map of figure 11. The black dot represents the robot, the red ellipse the uncertainty of our model, the blue one the standard MCL model . Both ellipses have been magnified 8x for better visibility.

The ground space model also refines the rotational part, at best with observed (true) edges in the room (see Figure 6). Please note that due to our ground space model a believed pose ”behind” the true pose has possibly the same likelihood as the true pose (assuming the ”behind” pose is not a wall or non explored area in the map and is not intersecting walls). This is due to the fact that both areas are not intersecting any walls. Such hypotheses can be removed by observing edges (see above) or passing a doorway to another room. In the case of MCL such hypotheses are automatically removed. Experiments show that it is not necessary to observe true wall data all the time. If no wall or edge is visible, the pose is tracked inside the *free space* of the map without loosing the true pose. The theoretical worst case is that the hypothesis covers the whole *free space* of the map, analogous to MCL with sonar sensors. In this case the situation is identical to the initial global self-localization.

## V. APPROXIMATION USING INTEGRAL IMAGES

The polynomial complexity of the reference implementation using a grid map is $O(n)$ where $n$ is the number of cells that need to be considered for the observation model. Depending on the implementation, additional cell lookups might be needed identifying the cells for considerations first. For instance computation of an area of $8m^2$ ($n = 3200$) needs 35ms on an 2.4 GHz Quad-Core processor with a cell size of 5cm of the map (for a single hypothesis !). The clear bottleneck is the piecewise lookup. Another issue is that it is hard to apply anytime constraints on the reference implementation.
Our approximated model describes the *ground space* by using square primitives. The main idea is that we start with the smallest enclosing square of the *ground space* and subtract smaller overlapping square areas that do not belong to the *ground space* itself. The area can be obtained in $O(1)$ using the *integral images* [17] technique. It calculates the exact area for any rectangular shape without loss of precision. Before we consider how to obtain the square primitives, we have a brief look on integral images.

### A. Integral Images

The integral image $I$ is an intermediate representation of the original image. It contains the sum of all gray scale pixel values of image $N$ with height $y$ and width $x$, i.e.,

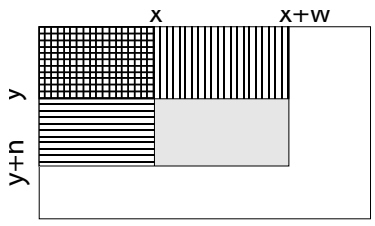



Fig. 7. Computation of values $F$ in the shaded region

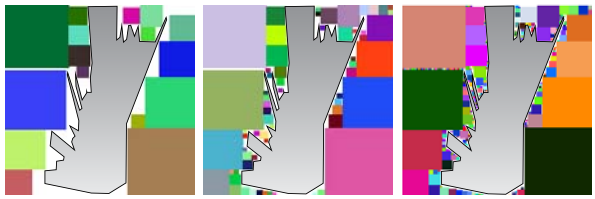
$$I(x,y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} N(x',y')$$

The integral image is computed recursively, by the formulas: $I(x,y) = I(x,y-1) + I(x-1,y) + N(x,y) - I(x-1,y-1)$ with $I(-1,y) = I(x,-1) = I(-1,-1) = 0$, therefore requiring only one scan over the input data. This intermediate representation $I(x,y)$ allows the computation of a rectangle value at $(x,y)$ with height and width $(h,w)$ using four references (see Fig. 7):

$$\begin{aligned} F(x,y,h,w) \quad = \quad & I(x,y) + I(x+w,y+h) - \\ & I(x,y+h) - I(x+w,y) \end{aligned}$$

We use an extension of the integral images proposed by Barczak [18]: It allows the rectangular area extraction at any rotational angle, but restricts the area to a square instead of the rectangle of the original implementation [17]. Please note that the original implementation of integral images allows only the use of rotated rectangles with angles of 0 and 45 degrees.

### B. Square Area Decomposition



(a) 16 Iterations (b) 64 Iterations (c) 256 Iterations

Fig. 8. Square Area Decomposition

The *ground space* is decomposed into a set of square primitives that are either used for adding or subtracting areas from the smallest enclosing square of the *ground space* (using the *Integral images*). In the computer graphics literature many solutions exist for this problem [19], [20], but unfortunately most of them are NP-hard for general rectangular areas, while heuristic with $O(n^2)$ exist for square primitives. The strategy for the square area decomposition works as follows (Figure 8):

1) Extract the maximum enclosing square $p_{rect}$ of the *ground space* area and create an empty bitmap with the dimension of $p_{rect}$.
2) Initialize the bitmap with an inverted *ground space* area bitmap
3) Extract the maximum square area of the *ground space*
4) Subtract the found square area pixels from the bitmap
5) go to 3, until $p_{maxiteration}$ squares are found or no more squares are found in the bitmap

The extracted squares are used to calculate the area of the *ground space* itself. The lookup for the maximum square area can be accelerated to $O(\log n)$ by using run-length encoding of the bitmap instead of accessing each pixel individually. In practice it is not really necessary to apply a full (zero error) decomposition. Figure 9 shows the decomposition error for $p_{maxiteration}$ iterations. One can see that the error dramatically decreases over all iterations. Experiments have shown that using 64 iterations is a good trade-off between speed and precision.

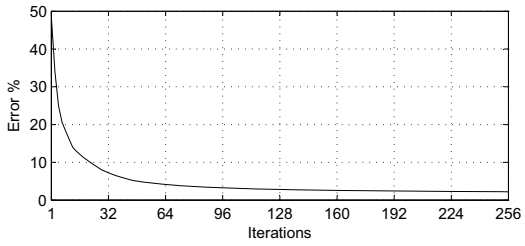



Fig. 9. Square area decomposition error

### C. Computational Consideration

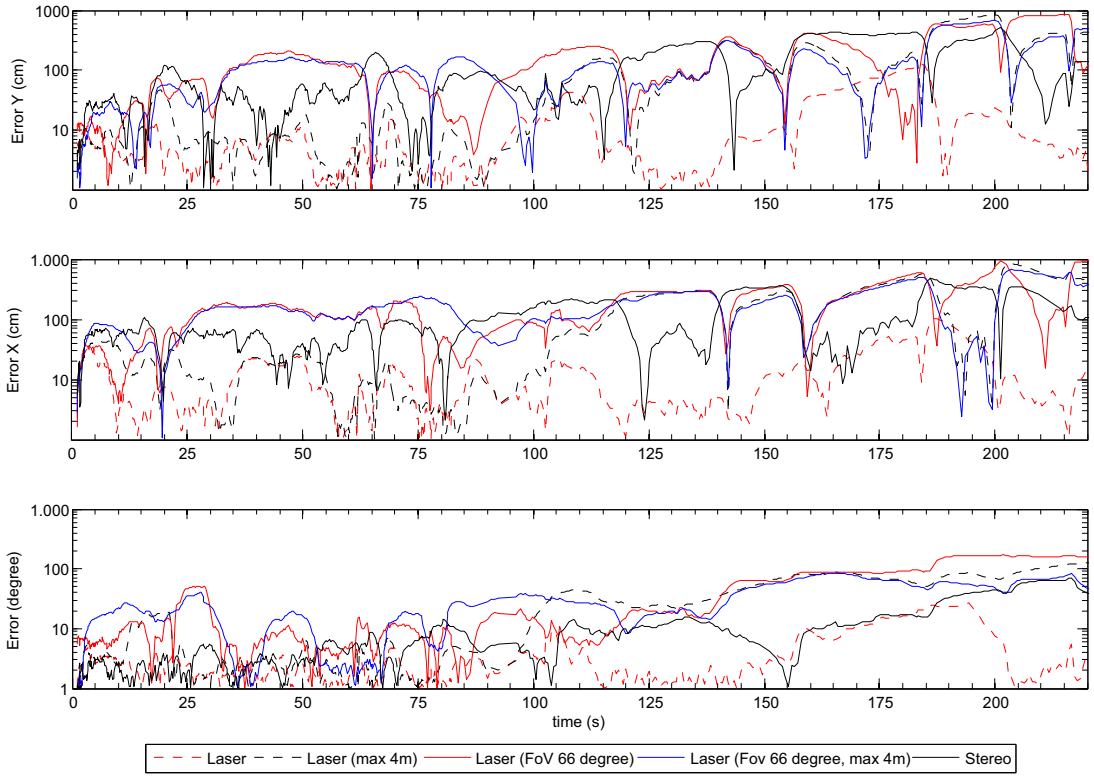
The polynomial complexity of the approximated model is $O(m^2 + \log n + mk)$ with $k$ hypothesis of MCL, $m$ iterations and $n$ cells cells of the *ground space*. $O(mk)$ is the number of lookups with the integral images for $k$ hypothesis. We simplify to $O(\log n)$, because $k,m$ are constant.

## VI. EXPERIMENTAL RESULTS

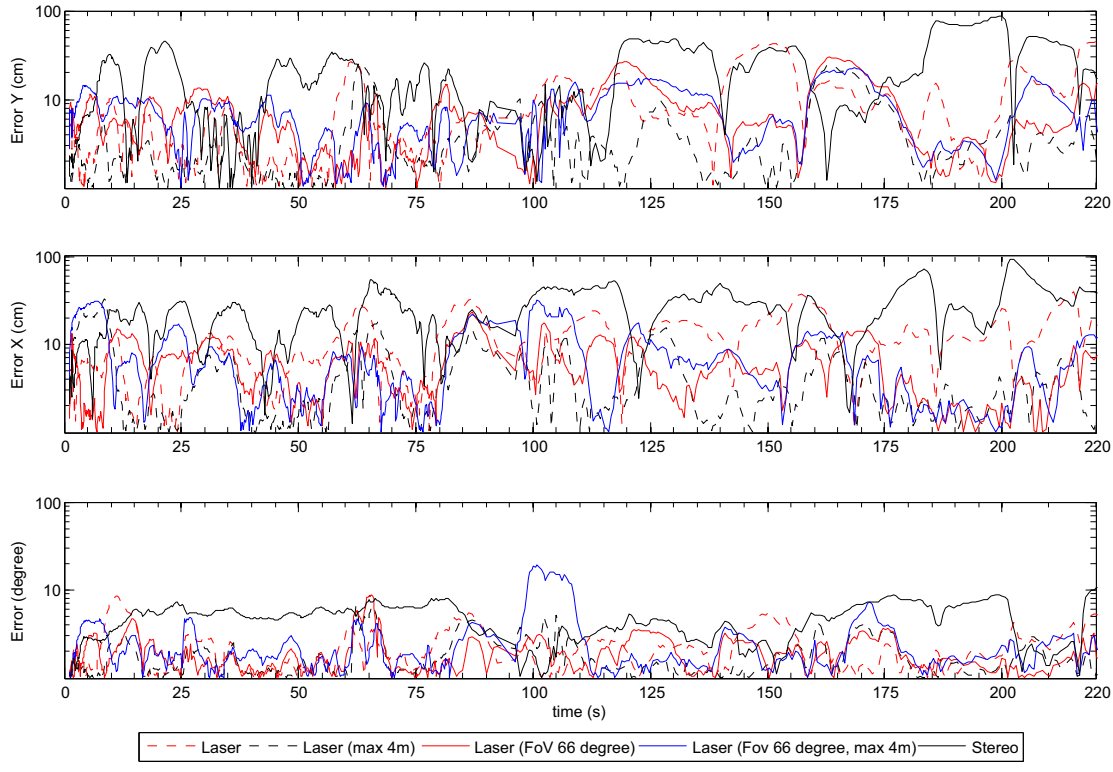
In this section we compare our proposed area-based observation model with the standard beam model using the well-known monte carlo localization method in practice. In order to demonstrate the high robustness of our model we use also stereo vision as sensor. The beam model with stereo uses an modified model [21] that also uses wall features. The main difference to the laser beam-model is a that it rejects outliers.

For our experiments we use a non-holonomic mobile robot manufactured by Bluebotics with an additional SICK LMS 200 laser range finder mounted to its front. We use the MOVEMENT prototype b/w stereo sensor [22] with vertical camera alignment and approx. 90 degrees field of view. The sensor is oriented into the robot's driving direction and mounted at a height of 30 cm over ground. We use the Videre-Design SRI stereo engine for dense stereo-data calculation at a resolution of 640x480 (VGA) and using 96 disparities.

We choose a typical office environment (see fig. 11) for data acquisition using the camera system, odometry and laser data. An external sensor system (multiple vision and laser scanners) for ground truth with a precision of $2cm$ is used. We use two different maps of the same environment: The first map contains the full environment like traditional localization systems [3], [21], [9]. The second map of the environment consists only of wall and ground segments like the footprint of a building. Please note that the furniture (in green) is only shown the sake of completeness but not included in the map. This is done on purpose to demonstrate the robustness and accuracy of our approach. The data of




(a) Standard MCL observation model




(b) Proposed observation model

Fig. 10. Average error of the robot localization of the tour as shown in figure 11. Please note that both plots have different scale.

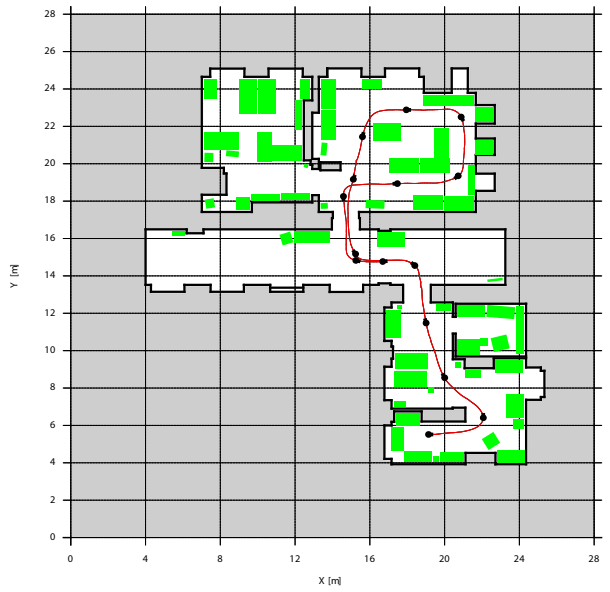



Fig. 11. Sample map of the used environment. The robot is shown as black circle/red line, green areas as furniture and walls as black . The furniture is not mapped.

TABLE I

AVERAGE LOCALIZATION ERROR OF THE COMPLETE DATA

| | MCL's Model | | Our Model | |
|---|---|---|---|---|
| Sensor | translation Error [cm] | rotation Error [deg] | translation Error [cm] | rotation Error [deg] |
| Laser / full range, complete Map | 1.825 | 1.176 | 2.135 | 1.023 |
| Laser / full range, incomplete Map | 3.864 | 4.776 | 2.253 | 1.383 |
| Laser / 4m range, incomplete Map | 23.396 | 31.589 | 1.825 | 1.041 |
| Laser / FoV 66 deg, incomplete Map | 32.975 | 47.166 | 1.994 | 1.175 |
| Laser / Fov 66 deg, 4m, incomplete Map | 25.190 | 21.716 | 2.736 | 2.750 |
| Stereo Vison full range, incomplete Map | 9.427 | 13.063 | 4.858 | 1.993 |

all sensors is recorded at 15 frames per seconds. All stereo data is calculated off-line from the previously recorded image pairs. We recorded representative eight tours through our lab with a total length of approximately 1600 meters. The robot moves with an average traveling speed of $0.65\frac{m}{s}$. In order to demonstrate the robustness and accuracy of our model, we additionally constrain the SICK laser scanner in its range of sensing and field of view.

Now let us consider one tour of the recorded data set in more detail: The robot starts at the desk at the bottom of the map and moves through the office in s-curves to the long corridor. After passing the corridor the robot enters a spacious office environment and moves through it. At the end, the robot leaves the spacious office and stops in the middle of the corridor.

Table I shows the average localization error on the complete data set. First we see that the unconstrained laser scanner with a complete map shows a good accuracy with the Beam-Model and our Model. Well, this result is not unexpected, MCL has already been proven a robust and precise method under these conditions. We want to emphasise that our proposed model works as good as the usual model under these conditions.

TABLE II

AVERAGE ERROR OF THE APPROXIMATED MODEL COMPARED TO THE REFERENCE IMPLEMENTATION

| Iterations | Translative Error [cm] | Rotative Error [deg] | Approximated Area [%] | execution time [ms] |
|---|---|---|---|---|
| 1 | +89.985 | +25.137 | 51.687 | 2 |
| 2 | +32.269 | +8.425 | 66.005 | 3 |
| 4 | +28.872 | +6.188 | 79.367 | 6 |
| 16 | +26.105 | +4.185 | 92.375 | 26 |
| 32 | +14.344 | +3.351 | 95.765 | 52 |
| 64 | +7.768 | +1.655 | 97.154 | 103 |
| 128 | +5.237 | +0.183 | 97.780 | 205 |
| 256 | +0.241 | +0.076 | 98.52 | 406 |
| reference implementation | - | - | 100.00 | 1421 |

The usage of the footprint map shows an influence on MCL while we still achieve a good accuracy compared to the unconstrained one. Figure 10(a) shows a plot of the pose error for the translative (X&Y, both relative to the robot heading) part and rotative part. The curve shows a max error of about 35cm before entering the large room. As expected the unmapped furniture degenerates the accuracy to a max error of 130cm, while the error of the angle shows a similar scheme. The constrained map has less influence on our model: The error shows a similarly smoothed curve. The better accuracy is due to the better handling of visibility information of the sensor data in contrast to the standard beam model. The more we constrain the laser data i.e. reducing the range, field of view or both, the more the accuracy degenerates with MCL. Again, this behavior is not unexpected. MCL relies on precise data in order to maintain the believed robot pose. With limited sensing, it has to rely on the motion model which leads to measured drift as shown in the plots. Our proposed model has less dependency on the constraint due to the fact that we are able to explicitly use visibility information. This helps to avoid the localization being stuck in local minima. For instance, constrained Laser with 66 degree field of view and max. 4m range is temporarily stuck in a local minimum at time index 160-170 while our model is not.

At last, we consider the accuracy of the approximated *ground space* using *integral images*. The results are shown in table II and in figure 12. We see that the rotaion error converges much faster than the translation error. This is due to the non-holonomic constraints of the robot we used.

## VII. CONCLUSION

In this paper we presented a new area-based observation model for Monte Carlo localization. The model shows good results in partially mapped environments where the traditional MCL fails or shows poor performance. We also demonstrated that the model is able to cope with a limited sensing range, limited field of view or even stereo vision. The approach performs as good as standard MCL under ordinary

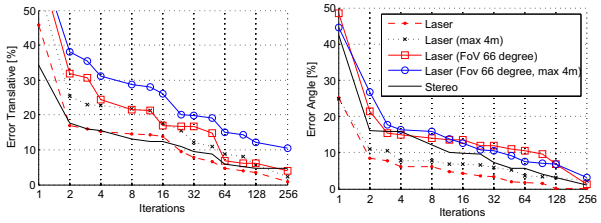



Fig. 12. Average Error of the Approximated Model compared to the reference implementation

conditions like a completely known environment. The approximation with *integral images* and image decomposition into squares provides a significant speed improvement with an adequate additional error. It should be emphasized that the area-based observation mode applies also to other sensor modalities such as ladar or sonar.

Our next steps will aim for an extended approximation model in 3D. Actually, the extension of *integral images* to 3D is straightforward, but preliminary experiments have shown that the extension from square to cubes decomposition of the volume is inefficient from the computational point of view: About 16x more iterations of the decomposition are needed to achieve the accuracy of the 2D Model.

## References


[1] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

[2] Kai O. Arras, Jose A. Castellanos, Martin Schilt, and Roland Siegwart. Feature-based multi-hypothesis localization and tracking using geometric constraints. *Robotics and Autonomous Systems*, 1(44):41–53, 2003.

[3] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge MA, first edition, 2005.

[4] B. Kalyan, A. Balasuriya, and S. Wijesoma. Multiple target tracking in underwater sonar images using particle-phd filter. pages 1–5, 2006.

[5] Emanuele Menegatti, Mauro Zoccarato, Enrico Pagello, and Hiroshi Ishiguro. Image-based monte carlo localisation with omnidirectional images. In *European Conference on Mobile Robots*, pages 17–30, 2004.

[6] Sebastian Thrun, Dieter Fox, and Wolfram Burgard. Monte carlo localization with mixture proposal distribution. In *AAAI/IAAI*, pages 859–865, 2000.

[7] Maren Bennewitz, Cyrill Stachniss, Sven Behnke, and Wolfram Burgard. Utilizing reflection properties of surfaces to improve mobile robot localization. In *IEEE International Conference on Robotics and Automation*, 2009.

[8] Patrick Pfaff, Cyrill Stachniss, Christian Plagemann, and Wolfram Burgard. Efficiently learning high-dimensional observation models for monte-carlo localization using gaussian mixtures. In *IIEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008, 2008.

[9] C. Plagemann, K. Kersting, P. Pfaff, and W. Burgard. Gaussian beam processes: A nonparametric bayesian measurement model for range finders. In *Robotics: Science and Systems (RSS)*, Atlanta, Georgia, USA, June 2007.

[10] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.

[11] R. Kümmerle, R. Triebel, P. Pfaff, and W. Burgard. Monte carlo localization in outdoor terrains using multilevel surface maps. *Journal of Field Robotics (JFR)*, 25:346–359, June - July 2008.

[12] Robert Sim and James J. Little. Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS), Beijing*, 2006.

[13] Andrew J. Davison, Ian Reid, Nicholas Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. In *IEEE Trans. PAMI*, 2007.

[14] S. Thrun, D. Fox, and W. Burgard. A real-time algorithm for mobile robot mapping with application to multi robot and 3d mapping. In *International Conference on Robotics & Automation*, San Francisco, CA, USA, 2000.

[15] P.Pfaff, C. Plagemann, and W. Burgard. Improved likelihood models for probabilistic localization based on range scans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2192–2197, 29 2007-Nov. 2 2007.

[16] R. I. Hartley A. and Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[17] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 2(57):137–154, 2004.

[18] Andre L. C. Barczak. Toward an efficient implementation of a rotation invariant detector using haar-like features. In *IVCNZ2005*, New Zealand, Dunedin, 2005.

[19] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics - Principles and Practice*, chapter 13. Second edition, 1997.

[20] Rober Sedgewick. *Algorithms in C*. first edition, 1992.

[21] P. Elinas and J.J. Little. omcl: Monte-carlo localization for mobile robots with stereo vision. In *Proceedings of Robotics: Science and Systems*, pages 373–380, Cambridge, MA, USA, 2005.

[22] P. Mayer, G. Edelmayer, G.J. Gelderblom, M. Vincze, P. Einramhof, M. Nuttin, T. Fuxreiter, and G. Kronreif. Movement -modular versatile mobility enhancement system. In *Procedings of IEEE International Conference on Robotics and Automation (ICRA), Rome*, 2007.