2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Probabilistic Mapping For Mobile Robots Using Spatial Correlation Models

Benjamin Pitzer
Research and Technology Center North America
Robert Bosch LLC, Palo Alto, CA, 94304
Email: benjamin.pitzer@us.bosch.com

Christoph Stiller
Institut für Mess- und Regelungstechnik
Universität Karlsruhe (TH), 76131 Karlsruhe, Germany
Email: stiller@mrt.uka.de

Abstract— **Generating accurate environment representations can significantly improve the autonomy of mobile robots. In this article we present a novel probabilistic technique for solving the full SLAM problem by jointly solving the data registration problem and the accurate reconstruction of the underlying geometry. The key idea of this paper is to incorporate spatial correlation models as prior knowledge on the map we seek to construct. We formulate the mapping problem as a maximum a-posteriori estimation comprising common probabilistic motion and sensor models as well as two spatial correlation models to guide the optimization. Instead of discarding data at an early stage, our algorithm makes use of all data available in the optimization process. When applied to SLAM, our method generates maps that closely resemble the real environment. We compare our approach to state-of-the-art algorithms, using both real and synthetic data sets.**

I. Introduction

A frequent issue in robot navigation is the concurrent exploration of metric maps while maintaining an accurate position estimate. If the robot were to have an a priori map, then localization would be a relatively easy task. Alternatively, if the robot were to have a precise, externally referenced position estimate, then mapping would simplify tremendously. However, problems in which the robot has no a priori map and no external position reference are particularly challenging. Such scenarios may arise for service robots inside and outside of buildings, AUVs, mining applications, or planetary surfaces. This problem has been referred to as either concurrent localization and mapping (CLM) or simultaneous localization and mapping (SLAM).

In this work we want to address the problem of reconstructing a map based on range measurements collected by a mobile platform. In recent years, building maps of physical environments with mobile robots has been a central problem in the robotics community. Research over the last decade has led to impressive results. Several successful algorithms emerged including Relaxation [1], SEIF [2], FastSLAM [3], MLR [4], TJTF [5], and Stochastic Gradient Descent [6]. Nearly all state-of-the-art methods are probabilistic and most of them are robust to noise and small variations of the environment. Comprehensive surveys can be found in [7] and more recently in [8].

A key limitation of almost all SLAM algorithms lies in the necessity to select appropriate landmarks. By reducing the sensor data to a representation by landmarks, a lot of the originally retrieved information is usually discarded. Another critical issue, which arises from using discrete landmarks in SLAM, is the problem of data association. Before fusing data into the map, new measurements are associated with existing map landmarks. This step has been proven crucial in practical SLAM implementations.

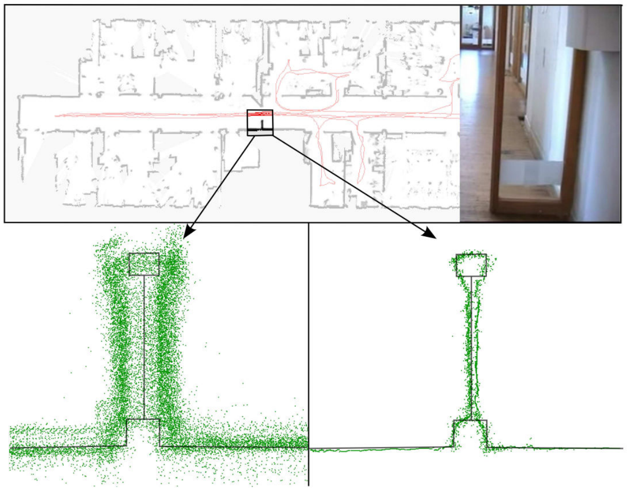


Fig. 1. The map of building 079 in Freiburg optimized with landmark-based SLAM (left) and by our algorithm using spatial correlation modells (right). Our algorithm achieves a significantly more accurate map reconstruction.

Due to the intrinsic limitations of sensor systems, spatial sensor interpretation is fundamentally an underconstrained problem. However, typical environments where current robots operate have some sort of structure hence measurements of this structure will be correlated. We believe that incorporating spatial correlation models as priors on the environment enables a robot to recover better world models. Creating an exact probabilistic model of all potential environments is not feasible and probably not even well defined; but in most cases, making basic assumptions is reasonable. For example, assuming the existence of locally smooth manifolds instead of randomly distributed surfaces is a reasonable model. To our knowledge, no algorithm for SLAM incorporates such knowledge in a probabilistic formulation.

We propose a novel formulation of the SLAM problem which incorporates spatial correlation models and does not

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

rely on the notion of landmarks. We show that this formulation is superior in cases where no salient landmark definition is feasible. Empirically we find that our formulation produces maps which closely resemble the environment.

II. Related Work

The topic of mappinig with mobile robots has been extensively studied. Despite significant progress in this area, it still poses great challenges. At present, we have robust methods for mapping environments that are static, structured, and of limited size. With the success of small-scale indoor and outdoor mapping, researchers are extending solutions to large scale environments [9], [10], [11] and 6 DoF poses [12]. To compare the capabilities of mapping algorithms, authors typically present generated maps, loop closures, and convergence behavior. Only recently Wulf et al. [13] presented a method to create ground truth pose data used to evaluate outdoor SLAM algorithms. Performance metrics demonstrating the accuracy of maps are practically nonexistent.

Early work in SLAM assumed that a map used for mobile robots could be modeled as a discrete set of landmarks. Different kinds of representations or maps have been proposed in robotics and in the artificial intelligence literature, ranging from low-level metric maps such as landmark maps [14] and occupancy grids [15], to topological graphs that contain high-level qualitative information [16], and even to multihierarchies of successively higher level abstractions [17].

Traditionally, SLAM implementations based on Kalman filter data fusion rely on simple geometric models for defining landmarks. This limits landmark based algorithms to environments suited to such models and tends to discard much potentially useful data. Only more recently, the work in [18] showed how to define landmarks composed of raw sensed data. A key component of our approach is recognizing that the typical landmark SLAM model assumes unstructured environments, i. e. environments with randomly independently distributed landmarks.

The occupancy grid framework, as proposed in [15] is used in many practical SLAM implementations [19], [20]. It employs a multidimensional (typically 2D or 3D) tessellation of space into cells, where each cell stores a probabilistic estimate of its occupancy state. For each grid cell, sensor measurements are integrated using Bayes rule to diminish the effect of sensor noise. This allows a variety of robotic tasks to be addressed through operations performed directly on the occupancy grid representation. The limited resolution of grid maps is the source for several problems. As pointed out in [21], a systematic error is introduced since the resolution of sensors typically used for perception varies with distance. Generally, the occupancy grid is modeled as a spatially uncorrelated random field. The individual cell states can be estimated as independent random variables. Again, a random structure is assumed for this model.

The assumption of line-based environments [22] and orthogonality as a geometrical constraints [23] have already been used by other researchers. Those approaches require features to be reliably extracted from the data as a preprocessing step which limits the performance of the subsequent SLAM algorithm.

From the discussion above we can identify some limitations of current SLAM approaches:

1) While much effort in robotic mapping is spent on large scale environments, little attention is put on the true accuracy of the resulting map.
2) Most map representations used in current SLAM approaches assume a random structure on the map or the features in the map. In fact, this is rarely the case, as all man-made environments are highly structured. In particular, the insides of buildings, a common workspace for mobile robots, are constructed with a well known methodology.
3) Information included in the sensor data is discarded at an early stage of processing: landmark maps discard much useful data while occupancy grid maps have problems with the inherent limited resolution.

These limitations motivate our research.

III. Formulation of the SLAM Problem

The following notation is adopted for a mathematical formulation of SLAM:

- $\mathbf{x}_t$: A vector describing the position and orientation of the robot at a time t.
- $\mathbf{u}_t$: The control vector that was applied at time $t-1$ and carries information about the change of the robot's pose.
- $\mathbf{z}_t^i$: The observations taken by the robot of the i^{th} feature.
- $\mathbf{c}_t$: A correspondence vector that contains a list of all features observed at time t.
- $\mathbf{m}$: A vector of map features $\mathbf{m} = \{\mathbf{m}_i\}$ representing the environment around the robot.

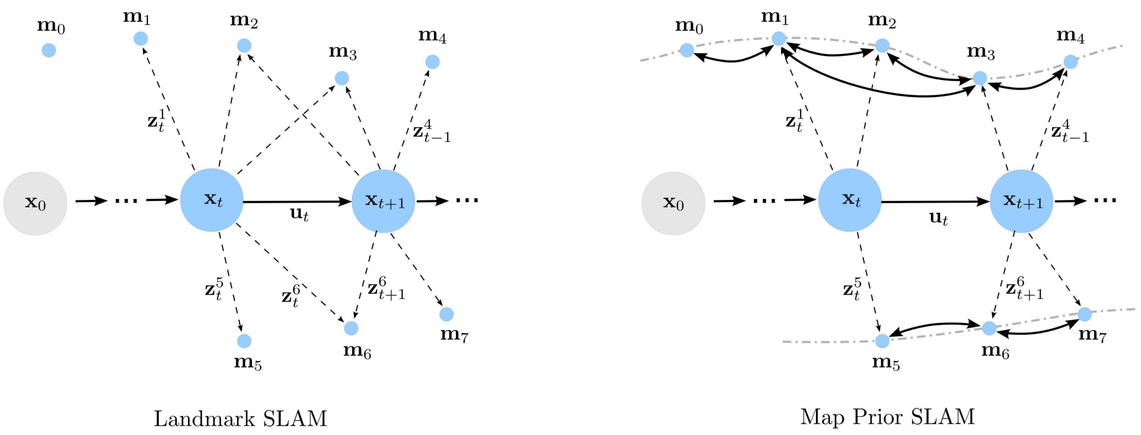
The goal of SLAM is to simultaneously estimate both the robot's pose and a map of its environment. In probabilistic SLAM this is often posed as Bayesian filtering formulation [7] in which one seeks to calculate the posterior over the robot's pose $\mathbf{x}_{1:t}$ along with the map $\mathbf{m}$:

$$p\left(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t}\right) \ . \tag{1}$$

Thrun et al. have shown in [11] that a closed form expression of this *full SLAM* posterior over all quantities can be obtained by recursively applying the Bayes rule and a subsequent induction over t:

$$\begin{aligned}\mathbf{p}\left(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t}\right) = \\ \eta\, p\left(\mathbf{x}_0\right) p\left(\mathbf{m}\right) \prod_t \left[p\left(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t\right) \prod_i p\left(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}, \mathbf{c}_t\right)\right]\end{aligned} \tag{2}$$

Here $p\left(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t\right)$ is known as the *motion model*, which describes state transitions of the robot's pose in terms of a probability distribution. The state transitions are assumed to follow a Markov process and are independent of both the observations and the map. The term $p\left(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}, \mathbf{c}_t\right)$



Landmark SLAM

Map Prior SLAM

Fig. 2. The left figure shows the structure of traditional landmark based SLAM algorithms. Each observation $\mathbf{z}_t^i$ is associated with a map feature $\mathbf{m}_i$. The right figure shows our approach which incorporates correlations between features into a probabilistic estimate. Such correlations are modeled as locally supported *map priors*.

denotes an *observation model*, which models an observation $\mathbf{z}_t^i$ from a known pose and a known map as a probability distribution. Both models have been studied well for a variety of robots and sensors. The two prior terms $p\left(\mathbf{x}_0\right)$ and $p\left(\mathbf{m}\right)$ characterize priors of the first robot pose and of the map respectively. Usually $p\left(\mathbf{x}_0\right)$ is used to anchor the initial pose to a fixed location. The map prior $p\left(\mathbf{m}\right)$ is typically assumed to be unknown and subsumed into the normalizer η. Finding the most probable solution to the full SLAM problem is the process of finding the set of poses $\hat{\mathbf{x}}_{1:t}$ and the map $\hat{\mathbf{m}}$ that maximizes the posterior probability of Eq. (2):

$$\hat{\mathbf{x}}_{1:t}, \hat{\mathbf{m}} = \underset{\mathbf{x}_{1:t}, \mathbf{m}}{\operatorname{argmax}}\, p\left(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}, \mathbf{c}_{1:t}\right) \ . \tag{3}$$

A graphical model of this formulation is presented on the left side of Fig. 2.

IV. SLAM WITH MAP PRIORS

In this contribution, we introduce prior expectations on typical environments into SLAM by means of suitable a priori distributions $p(\mathbf{m})$. First we want to eliminate the notion of landmarks. In the previous formulation we assumed that the correspondences $\mathbf{c}_t$ are known beforehand which enables us to uniquely assign a landmark $\mathbf{m}_i$ to each observed feature. However, in practical SLAM implementations correspondence assignment becomes a demanding task. In general, correspondence between measurements taken at different time instances are non-unique and the imposture thereof is a main source of deteriorated results for many SLAM implementations. In our formulation, we consider the measurements directly without extraction of any landmarks. Instead, we claim there are no existing immediate correspondences between measurements. In fact, this claim is quite reasonable for a number of situations. For example, a mobile robot equipped with a lidar, which takes a finite number of measurements while it is in motion, is very unlikely to measure the exact same spot twice.

Here are the key modifications to the original SLAM formulation:

1) Each observation $\mathbf{z}_t^i$ creates a new feature in the map.
2) We assume no correspondences between observations and known features.
3) Instead of correspondences, we use correlation models as a map prior $p\left(\mathbf{m}\right)$ to guide the estimation of the robot's pose and the map.

The new posterior for this formulation is:

$$p\left(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}\right) = \tag{4}$$

$$\eta\, p\left(\mathbf{x}_0\right) p\left(\mathbf{m}\right) \prod_t \left[p\left(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t\right) \prod_i p\left(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}\right) \right]$$

A graphical model of this new formulation is presented in the right side of Fig. 2.

Our modifications have some interesting implications. First, the state space of our optimization problem will be significantly larger than landmark based approaches because of the one-to-one correspondence of measurements and map features. It is important to realize that for the optimization of Eq. (4) a good map prior is vital. The observation model, the motion model, as well as the prior of the first pose considered independently are explained best by the measurements themselfs. Without any map prior the most probable solution of Eq. (4) would be the measurement itself. An optimization Eq. (4) will move points locally to comply with the map prior model. This is fundamentally different from ICP-style rigid alignment techniques where only the robot pose is optimized. The point motion will be constraint due to the dependence of measurement and pose. In fact, a movement of a point will create a counter potential for the point and for the corresponding pose to comply with the measurement model. In other words, maximizing the posterior probability Eq. (4) will lead to a set of poses and map features that best explain the measurements as well as the prior model.

V. PROBABILISTIC MODELS

The robot's motion is represented using a common probabilistic motion model where the robot is assumed to perform

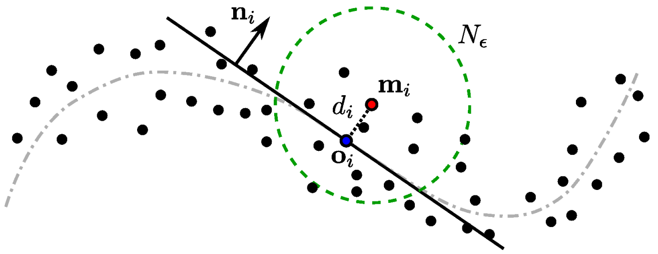


Fig. 3. The *manifold model* uses a fixed neighborhood N_ε of points to create a tangent line defined by a point $\mathbf{o}_i$ and the normal $\mathbf{n}_i$. The potential is then modeled as a Gaussian-type function over the projected distance d to the tangent line.

a series of a rotation, a translation, and a second rotation [24]. Observations are modeled as a range measurement along a beam, which originates at the local coordinate system of the sensor [24]. The prior term $p(\mathbf{x}_0)$ *anchors* the first pose at its position (e. g. origin of the global coordinate system) which is easily expressed by a Gaussian-type distribution using a very small standard deviation.

VI. Spatial Correlation Models

The probability distribution $p(\mathbf{m})$ in Eq. (4) represents a *prior distribution* of all measured scenes. An exact probabilistic model of this distribution is not feasible and probably not even well defined. Hence we focus on partial models, which represent properties of the surface structure. In our approach, we use locally defined spatial correlation models representing two properties: manifoldness $f_M(\mathbf{m})$ and the orientation $f_O(\mathbf{m})$. The final prior $p(\mathbf{m})$ is defined as:

$$p(\mathbf{m}) = \frac{1}{\eta} f_M(\mathbf{m}) f_O(\mathbf{m}) \ . \tag{5}$$

In order to make this expression a valid probability density function, we introduce a constant factor η, which is the integral over all other factors and, therefore, normalizes $p(\mathbf{m})$ to be a probability density function. In practice, this factor can be safely ommited as the normalization does not have an effect on the optimization of Eq. (4).

A. Manifold Model

The intuition of this correlation model is that map observations belong to structured surfaces in the robot's environment. This means that for a 2D map the most probable surface must be a compact, locally connected, one-dimensional manifold, possibly with boundary, and embedded in $\mathbb{R}^2$. The first step towards defining a potential function which captures this property is to compute a tangent line associated with each map point $\mathbf{m}_i$. A tangent line is defined by a point $\mathbf{o}_i$ and normal $\mathbf{n}_i$. For each point, we choose a local neighborhood N_ε of variable diameter (typically $\varepsilon = 10 \dots 50$ points). The center $\mathbf{o}_i$ is taken to be the centroid of N_ε, and the normal $\mathbf{n}_i$ is determined using principal component analysis [25]: the eigenvector with the smallest eigenvalue corresponds to the normal $\mathbf{n}_i$. The projected distance d_i of the point onto its tangent line is defined by the dot product:

$$d_i = (\mathbf{m}_i - \mathbf{o}_i) \cdot \mathbf{n}_i. \tag{6}$$

Now, we define a Gaussian-type manifold potential function of the form:

$$f_M(\mathbf{m}) = \prod_i \exp\left\{-\frac{d_i^2}{2\sigma_M}\right\}, \tag{7}$$

where σ_M is the variance of tangent line distances. Fig. 3 illustrates the properties of this correlation model. The data points are drawn to their corresponding tangent lines. Hence the most probable arrangement of map points regarding this potential is when all points are located on a one-dimensional manifold. It should be noted that the manifold potential is a set of locally supported functions. The size of N_ε defines the region of influence. We allow ε to adapt locally which makes less stringent the requirement that the data is uniformly distributed over the surface. To select and adapt the neighborhood size, we use a kernel density estimator [26] and set ε proportional to the computed density.

B. Shape Model

The shape model addresses consistency of surface orientations. If two surface regions (cf. Fig. 4) belong to the same physical surface, the orientation of edges representing the same portion should have a consistent orientation. In other words, we are looking for geometric relations (parallelism, orthogonality) of adjacent surface regions since we assume a predominantly rectilinear environment.

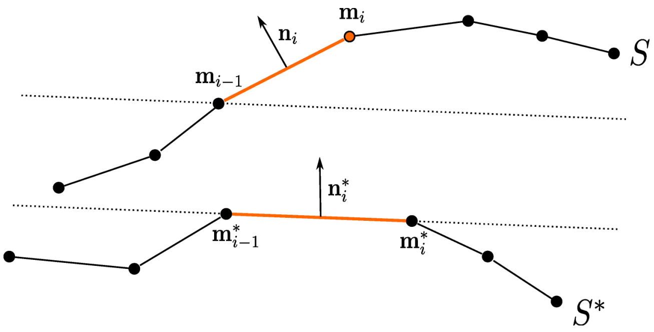


Fig. 4. The *shape potential* uses the orientation of adjacent surface regions. The differences between two corresponding normals $\mathbf{n}_i$ and $\mathbf{n}_i^*$ are modeled as Gaussian-type functions over the normals differences.

A simple approach is to examine the normals of adjacent surface regions. If the angle is close to one of 0°, 90°, 180°, or 270° the shape potential will draw the points towards a rectilinearity case. Such a potential can be defined as follows: Let $\mathbf{m}_i$ and $\mathbf{m}_{i-1}$ be neighbors on the same surface region. The normal on the surface region is defined by:

$$\mathbf{n}_i = \frac{(\mathbf{m}_{i-1} - \mathbf{m}_i)^\perp}{||\mathbf{m}_{i-1} - \mathbf{m}_i||} = \underbrace{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}}_{:=M_\perp} \frac{(\mathbf{m}_{i-1} - \mathbf{m}_i)}{||\mathbf{m}_{i-1} - \mathbf{m}_i||} \ . \tag{8}$$

Now, let $\mathbf{n}_i$ and $\mathbf{n}_i^*$ be the normals on two corresponding surface regions S and S^* respectively (cf. Fig. 4). Then the shape potential for the 0° case has the form of a quadratic normal distribution:

$$f_{0^\circ}(\mathbf{m}) = \exp\left\{-\frac{1}{2}(\mathbf{n}_i^* - \mathbf{n}_i)^T \Omega_O (\mathbf{n}_i^* - \mathbf{n}_i)\right\} \tag{9}$$

Here Ω_O corresponds to a covariance matrix for the orientation of adjacent surface regions. In a similar way, we define individual functions for the 90°, the 180°, and the 270° case. We also allow for a small margin of 10° within we assign paired regions to the categories. All other cases that do not fall into one of the geometric relations will not contribute to the shape potential.

VII. Implementation

In Eq. (4) we defined a novel probabilistic model for the full SLAM problem; and in the previous sections, we discussed the different components of this model. All map features and robot poses are calculated simultaneously from the *maximum a-posteriori solution (MAP)* of Eq. (4). Now we want to focus on a practical implementation to calculate this solution.

Since maxima of Eq. (4) are unaffected by monotone transformations, we can take the negative logarithm of this expression to turn it into a sum and optimize this expression instead:

$$\begin{aligned}
\hat{\mathbf{x}}_{1:t}, \hat{\mathbf{m}} &= \operatorname*{argmin}_{\mathbf{x}_{1:t},\mathbf{m}} p\left(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}\right) \\
&= \operatorname*{argmin}_{\mathbf{x}_{1:t},\mathbf{m}} - \log \eta - \log p\left(\mathbf{x}_0\right) - \log p\left(\mathbf{m}\right) \\
&\quad - \sum_t \log p\left(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t\right) \\
&\quad - \sum_t \sum_i \log p\left(\mathbf{z}_t^i | \mathbf{x}_t, \mathbf{m}_i\right) \\
&= \operatorname*{argmin}_{\mathbf{x}_{1:t},\mathbf{m}} E\left(\mathbf{x}_{0:t}, \mathbf{m}\right)
\end{aligned} \tag{10}$$

Finding the most probable solution reduces now to finding the global minimum of the function $E\left(\mathbf{x}_{0:t}, \mathbf{m}\right)$, which is a sum of log-likelihoods. The term $-\log \eta$ is a constant normalization factor and is not relevant for minimizing $E\left(\mathbf{x}_{0:t}, \mathbf{m}\right)$.

Our algorithm consists of three main steps: first we use the *motion model* and the *observation model* to calculate an initial estimate for $\mathbf{x}_{1:t}$ and $\mathbf{m}_{1:i}$ respectively. Next, an initialization is performed to improve this estimate. Finally, we use a non-linear conjugate gradient variant to find the parameters which minimize $E\left(\mathbf{x}_{0:t}, \mathbf{m}\right)$. An outline if this algorithm is presented in Alg. 1.

A. Initialization

Unfortunately, the objective function $E\left(\mathbf{x}_{0:t}, \mathbf{m}\right)$ is highly non-linear and thus finding the global minimum is known to be difficult. For this reason, we use a scan alignment algorithm prior to the optimization. In particular, we use the well known *iterative-closest-point* (ICP) algorithm [27] to create an initial alignment and, therefore, a better starting point for our optimization. Our experiments show that this starting point is usually sufficiently close to the global minimum of E that the following optimization procedure will converge into the correct solution. The details of the ICP algorithm are omitted for brevity, but can be found in contemporary texts [28].

Algorithm 1 Calculate $\hat{\mathbf{x}}_{1:t}, \hat{\mathbf{m}}$

```
for all controls $\mathbf{u}_t$ do
    $\mathbf{x}_t \leftarrow$ motion_model $(\mathbf{u}_t, \mathbf{x}_{t-1})$
    for all observations $\mathbf{z}_t^i$ do
        $\mathbf{m}_i \leftarrow$ observation_model $(\mathbf{x}_t, \mathbf{z}_t^i)$
    end for
end for
$\mathbf{x}_{1:t}, \mathbf{m} \leftarrow$ iterative_closest_point $(\mathbf{x}_{1:t}, \mathbf{m})$
repeat
    create prior model $p(\mathbf{m})$
    fix state variables $\mathbf{x}_{1:t}$
    $\mathbf{m} \leftarrow$ conjugate_gradient_iteration $(\mathbf{x}_{1:t}, \mathbf{m})$
    fix state variables $\mathbf{m}_{1:i}$
    $\mathbf{x}_{1:t} \leftarrow$ conjugate_gradient_iteration $(\mathbf{x}_{1:t}, \mathbf{m})$
until convergence
```

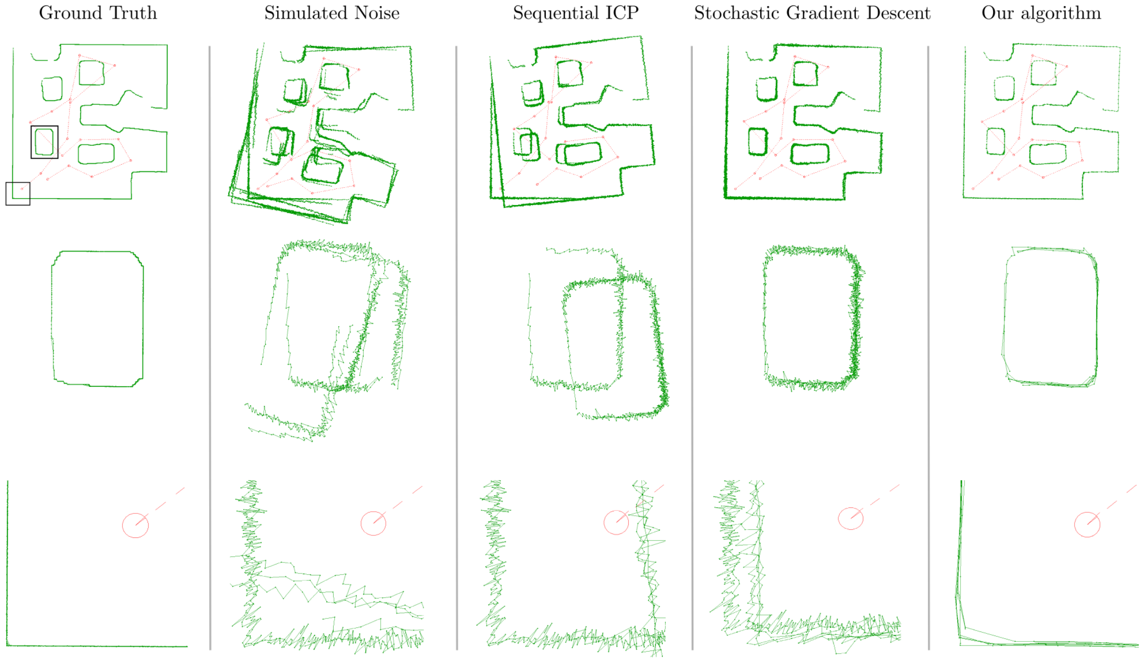
B. Optimizing Using Conjugate Gradient

The most probable path and the most probable map results from finding the global minimum of the function $E\left(\mathbf{x}_{1:t}, \mathbf{m}\right)$. The minimization itself is a high dimensional and sparse optimization problem. Therefore, the *Nonlinear Conjugate Gradient Method* (CG) is used to find a good solution. A detailed description of this a method can be found in [29]. In our implementation we use a modified version of CG which addresses the structure of our state space. Each CG iteration consists of two sub-optimization steps. First, we fix all pose state variables and optimize the map feature positions. Next, the map features are fixed and the positions are optimized. By splitting the optimization in two steps, we loose the optimality of our solution; however, in practice, this scheme lead to a better convergence than optimizing all state variables simultaneously. Each sub-optimization step employs a standard Newton-Raphson line search and the Fletcher-Reeves formulation to linearly combine the negative gradient of the function $E\left(\mathbf{x}_{0:t}, \mathbf{m}\right)$ with previous such gradients. A detailed outline of this procedure is presented in Alg. 1. The derivation of the gradients of $E\left(\mathbf{x}_{1:t}, \mathbf{m}\right)$ is presented in the appendix.

VIII. Experimental Results

A. Synthetic Data

To study the properties of our approach, we use synthetic data created in a simulated environment [30]. The dataset consists of 16 simulated 360° scans with 720 measurements in each scan. The ground truth of the whole set and of two details are presented on the left side of Fig. 5. This ground truth was then distorted by adding Gaussian noise to the measurements (range and bearing) and to the odometry in order to create a realistic input for all algorithms. For a comparison, we use a variant of the ICP algorithm [27], which incrementally registers all scans and the state-of-the-art *Stochastic Gradient Descent (SG)* optimizer by Olson et. al [6]. The thickness of walls is a general indication of the error distribution—a divergence of the robot pose typically results in map distortions like bend/double walls while noise



Ground Truth

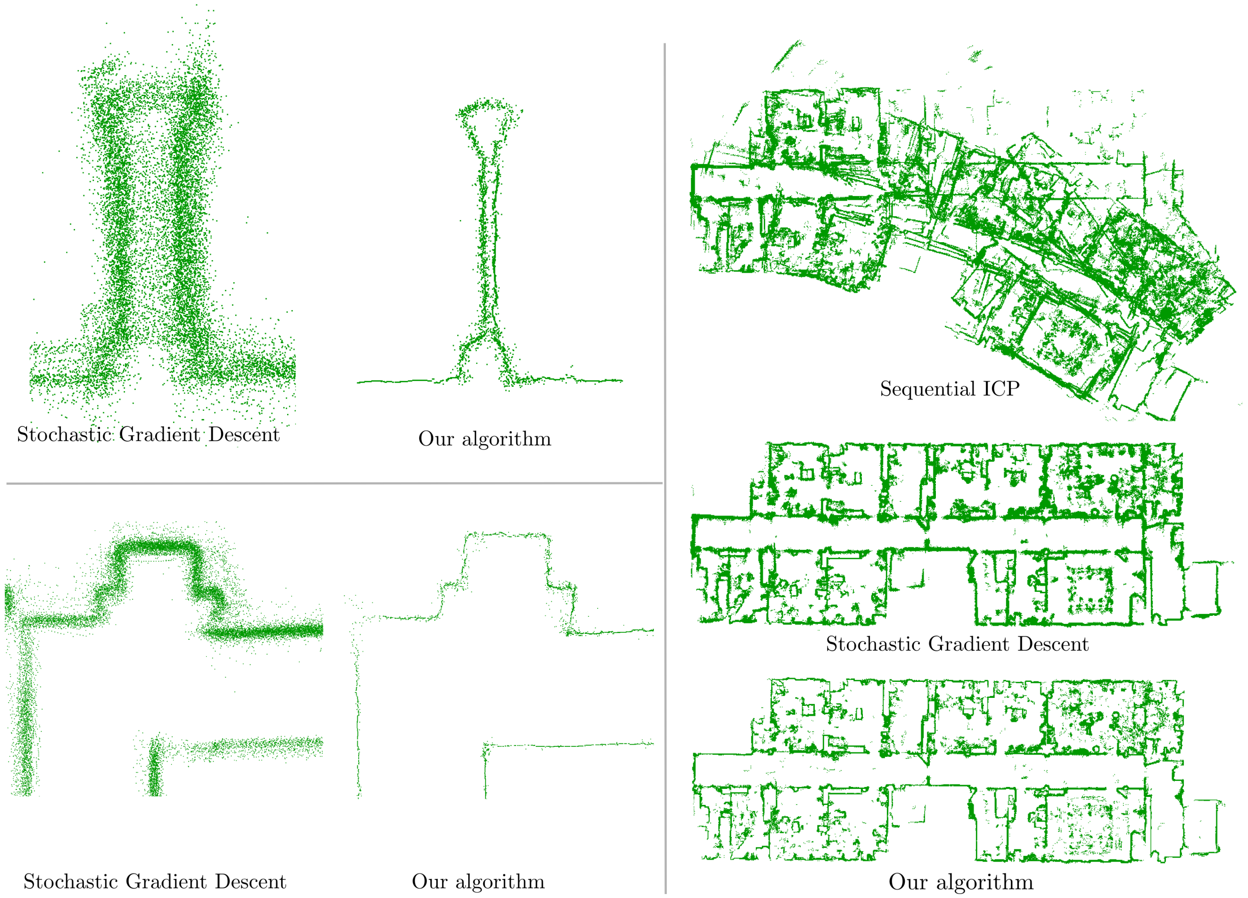
Simulated Noise

Sequential ICP

Stochastic Gradient Descent

Our algorithm

Fig. 5. Evaluation of our algorithm on a synthetic dataset. The first row shows the full dataset, whereas the second and third row depicts magnified details. The ICP algorithm [27] yields a good pairwise alignment but results in an inconsistent map (middle). Olson's Stochastic Gradient Descent [6] algorithm produces a consistent map on the first glance, but the details reveal fuzzy surfaces (first detail) and small pose error residuals (second detail). Our algorithm creates a globally consistent map and is able to eliminated measurement errors almost completely (right).



Stochastic Gradient Descent

Our algorithm

Stochastic Gradient Descent

Our algorithm

Sequential ICP

Stochastic Gradient Descent

Our algorithm

Fig. 6. A comparison of our algorithm to sequential registration with the ICP algorithm (Besl et al. [27]) and Stochastic Gradient Descent (Olson et al. [6]) on the fr079 dataset. The right side shows the full dataset and on the left side we present magnifications of two details.

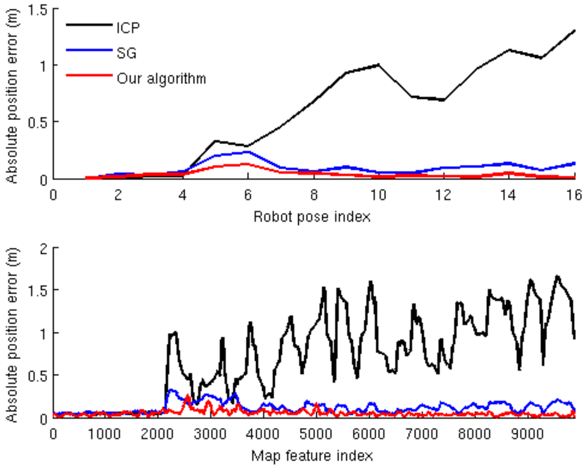


Fig. 7. Absolute pose and map feature errors for the synthetic dataset.

of the range sensor has a fixed mean and makes walls appear fuzzy and blurred (see Fig. 5). We also assess the algorithm performance by comparing the reconstructed trajectory and the reconstructed map with the available ground truth (see Fig. 7). On the one hand, ICP is able to align groups of scans correctly, but it fails to create a globally consistent map. This behavior is expected since ICP is a pairwise alignment algorithm. On the other hand, SG produces a consistent map, although some residual pose errors remain. Both ICP and SG adjust robot poses only and therefore feature measurement errors are not corrected. In contrast, our algorithm creates a globally consistent map and is able to eliminate measurement errors. Fig. 7 shows the absolute robot pose and map feature errors and the following table presents the corresponding first and second moments:

	Pose error (m)		Map error (m)	
Algorithm	mean	var	mean	var
ICP	0.597	0.199	0.702	0.238
SG	0.088	0.115	0.115	0.005
Our algorithm	0.035	0.048	0.047	0.002

B. Real Data

In the second experiment we use data gathered by a real robot. Here, we use the fr079 data set which is publicly available at the Robotics Data Set Repository (Radish) [31]. It consists of 4791 scans from a Sick LMS lidar with odometry pose estimate for each scan. For a comparison we again use Olson's Stochastic Gradient optimizer [6]. The reconstruction of the full map is presented on the right side of Fig. 6. Both SG and our algorithm result in a similar map. One may notice that the walls appear thinner in the map reconstructed by our algorithm, which quantitatively shows our algorithm provides a sensible estimation of the robot path. The details on the left side of Fig. 6 reveal a significantly better registration of the data. Our algorithm performs very well on straight walls since those features are represented best by our correlation models. Some outliers and smoothed corners are produced on sharp features since our models are not well defined at corners.

IX. Conclusion

This paper has presented a novel approach for probabilistic mapping. We used a map representation which stores all observations as unique features in this map. Instead of assuming correspondences between observations, we incorporated two spatial correlation models as map priors to guide the optimization. With this approach, we formulated the full SLAM problem as a maximum a-posteriori estimation problem which we optimized using a nonlinear conjugate gradient method. Finally, we demonstrated a practical algorithm that has been evaluated on synthetic data and on a large dataset available to the public.

In this particular research, we focused on using correlation models to guide the optimization. This is fundamentally different from the notion of landmarks and landmark correspondences used in traditional SLAM implementations. The next logical step is a combination of correlation models with landmark correspondences and will be subject of our further research.

A potential drawback of our approach is that the optimization may get stuck in a local minimum, which is a universal problem of non-convex optimizations. A sufficient initialization, as presented earlier, usually leads to good results. However, it remains unclear whether this is a substantial shortcoming of our approach.

This paper focused on mapping in planar environments with 2D maps and 3 DoF poses to be estimated. Nevertheless, we believe that a similar approach is feasible for 3D maps and 6 DoF poses.

Acknowledgment

The synthetic dataset was created by using the Player/Stage System [30]. The fr079 data set was obtained from the Robotics Data Set Repository (Radish) [31]. Thanks go to Cyrill Stachniss for providing this data.

Appendix: Log-posterior Gradients

For notational brevity we write:

$$\Theta\left(\mathbf{m}\right) \quad = \quad const. - \log p\left(\mathbf{m}\right) \; . \qquad (11)$$

The gradients for the log-prior model can be derived by calculating the partial derivatives of $\Theta\left(\mathbf{m}\right)$ with respect to $\mathbf{m}_i$:

$$\frac{\partial \Theta_M\left(\mathbf{m}\right)}{\partial \mathbf{m}_i} \quad = \quad \frac{1}{\sigma_M} \cdot \left(\mathbf{m}_i - \mathbf{o}_i\right) \; , \qquad (12)$$

$$\frac{\partial \Theta_O\left(\mathbf{m}\right)}{\partial \mathbf{m}_i} \quad = \quad -\Omega_O\left(\mathbf{n}_i^* - \mathbf{n}_i\right) \frac{\partial \mathbf{n}_i}{\partial \mathbf{m}_i} \; , \qquad (13)$$

with the derivative of a normal with respect to $\mathbf{m}_i$:

$$\frac{\partial \mathbf{n}_i}{\partial \mathbf{m}_i} = M_\perp \left[\frac{I}{\left\|\mathbf{m}_{i-1} - \mathbf{m}_i\right\|_2} - \frac{\left(\mathbf{m}_{i-1} - \mathbf{m}_i\right)\left(\mathbf{m}_{i-1} - \mathbf{m}_i\right)^{\mathrm{T}}}{\left\|\mathbf{m}_{i-1} - \mathbf{m}_i\right\|_2^3} \right] .$$

References

[1] T. Duckett, S. R. Marsland, and J. L. Shapiro, “Fast, on-line learning of globally consistent maps,” *Autonomous Robots*, vol. 12, no. 3, pp. 287–300, May 2002.

[2] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *International Journal of Robotics Research*, 2004.

[3] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “Fastslam: a factored solution to the simultaneous localization and mapping problem,” in *18th National Conf. on Artificial Intelligence*, 2002, pp. 593–598.

[4] U. Frese and T. Duckett, “A multigrid approach for accelerating relaxation-based SLAM,” in *Proc. IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR 2003)*, Acapulco, Mexico, 2003, pp. 39–46.

[5] M. A. Paskin, “Thin junction tree filters for simultaneous localization and mapping.” in *18th International Joint Conf. on Artificial Intelligence*, 2003, pp. 1157–1166.

[6] E. Olson, J. Leonard, and S. Teller, “Fast iterative alignment of pose graphs with poor initial estimates,” in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, May 15-19 2006, pp. 2262–2269.

[7] S. Thrun, “Robotic mapping: A survey,” in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[8] T. Bailey and H. Durrant-Whyte, “Simultaneous localisation and mapping (slam): Part i - state of the art,” *Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, June 2006.

[9] M. Bosse, P. M. Newman, J. J. Leonard, M. Soika, W. Feiten, and S. J. Teller, “An atlas framework for scalable mapping.” in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 14-19 2003, pp. 1899–1906.

[10] K. Konolige, “Large-scale map-making,” in *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*, D. L. McGuinness and G. Ferguson, Eds. AAAI Press / The MIT Press, 2004, pp. 457–463.

[11] S. Thrun and M. Montemerlo, “The GraphSLAM algorithm with applications to large-scale mapping of urban structures,” *International Journal on Robotics Research*, vol. 25, no. 5/6, pp. 403–430, 2005.

[12] A. Nüchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun, “6d slam with application in autonomous mine mapping,” in *Proceedings IEEE 2004 International Conference Robotics and Automation (ICRA 2004)*, New Orleans, USA, April 2004, pp. 1998 – 2003.

[13] O. Wulf, A. Nuchter, J. Hertzberg, and B. Wagner, “Ground truth evaluation of large urban 6d slam,” *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 650–657, 29 2007-Nov. 2 2007.

[14] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (slam) problem,” *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 3, pp. 229–241, Jun 2001.

[15] A. Elfes, “Using occupancy grids for mobile robot perception and navigation,” *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[16] B. Kuipers and Y.-T. Byun, “A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations,” pp. 47–63, 1993.

[17] J.-A. Fernandez-Madrigal and J. Gonzalez, “Multihierarchical graph search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 103–113, 2002.

[18] J. Nieto, T. Bailey, and E. Nebot, “Recursive scan-matching slam,” *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 39–49, January 2007.

[19] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, “A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements,” in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[20] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, “Fast and accurate slam with rao-blackwellized particle filters,” *Robot. Auton. Syst.*, vol. 55, no. 1, pp. 30–38, 2007.

[21] M. Montemerlo and S. Thrun, “A multi-resolution pyramid for outdoor robot terrain perception,” in *Proceedings of the AAAI National Conference on Artificial Intelligence*. San Jose, CA: AAAI, 2004.

[22] A. Garulli, A. Giannitrapani, A. Rossi, and A. Vicino, “Mobile robot slam for line-based environment representation,” in *44th IEEE European Control Conference on Decision and Control*, 2005, pp. 2041–2046.

[23] A. Harati and R. Siegwart, “Orthogonal 3d-slam for indoor environments using right angle corners,” in *3rd European Conference on Mobile Robotics*, 2007.

[24] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[25] H. Hoppe, T. Derose, T. Duchamp, J. Mcdonald, and W. Stuetzle, “Surface reconstruction from unorganized points,” *Computer Graphics*, vol. 26, no. 2, pp. 71–78, 1992.

[26] E. Parzen, “On estimation of a probability density function and mode,” *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[27] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes.” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, no. 2, pp. 239–256, 1992.

[28] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” in *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.

[29] J. R. Shewchuk, “An introduction to the conjugate gradient method without the agonizing pain,” August 1994. [Online]. Available: http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf

[30] B. Gerkey, R. Vaughan, and A. Howard, “The player/stage project: Tools for multi-robot and distributed sensor systems,” in *11th International Conference on Advanced Robotics (ICAR 2003)*, Coimbra, Portugal, jun 2003. [Online]. Available: citeseer.ist.psu.edu/gerkey03playerstage.html

[31] A. Howard and N. Roy, “The robotics data set repository (radish),” 2003. [Online]. Available: http://radish.sourceforge.net/