DUAL OF THE FACTORED SOLUTION TO THE SIMULTANEOUS LOCALIZATION AND MAPPING PROBLEM

D. Rodriguez-Losada, P. San Segundo, F. Matia, R. Galan, A. Jiménez, L. Pedraza.

Technical University of Madrid, Spain
C/ Jose Gutierrez Abascal 2, 28006 Madrid.
diego.rlosada@upm.es

Abstract: The SLAM estimation problem has an interesting factorization that decouples the path estimation and the map building problem, using a particle filter to estimate over the possible robot paths. Although there are several successful approaches to this idea, there is a lack of application to indoor feature based mapping. This paper presents a novel factorization, which is the dual of the existing one, that decouples the SLAM into a map estimation and a localization problem, using a particle filter to estimate over maps. We have implemented and tested this formulation, successfully building feature based maps of indoor environments with particle filters for (up to our knowledge) the first time.

Keywords: SLAM, particle filter, feature based maps

1. INTRODUCTION

Reliable localization, which is commonly based on an internal representation (map) of the environment, is a key component of any autonomous robot. Although supplying an a priori man-made map is sometimes possible, being able to automatically build a map of the environment with a mobile robot possesses great advantages for many applications.

The probabilistic approach has dominated the solution to the Simultaneous Localization and Mapping (SLAM) problem, which is considered the key for developing a truly autonomous mobile robot. Many different approaches exist: occupancy grid mapping, Expectancy Maximization, hybrid approaches and even topological SLAM. See (Thrun, 2002) for a survey on probabilistic robotic mapping.

Since the seminal paper (Smith, *et al.* 1988), the use of an Extended Kalman Filter (EKF), has probably been the most extended approach to stochastic mapping. One of its drawbacks is its computational cost, that grows proportionally to the square of the number 'n' of map objects, O(n^2), which limits the algorithm real time application to maps with a few hundreds of features. Nevertheless, much recent research has shown that this computational cost can be reduced even without loosing the optimality of the filter (Williams et al 2002; Guivant and Nebot, 2001). Also, the EKF inverse formulation, known as the Information Filter has been recently successfully used with a great reduction of the computational cost. The consistency and convergence properties proofs of the SLAM-EKF algorithm of (Dissanayake et al, 2001) are known to be the strongest proofs of convergence in the SLAM domain, but it is a known problem that linearizations of the EKF can lead to divergence of the filter but only (Julier and Uhlmann, 2001; Castellanos et al. 2004) have pointed out this effect on the basic algorithm, while (Rodriguez-Losada et al. 2006) deals with the inconsistency problem when applying the use of local maps to reduce the computational cost. Thus these methods also have problems for their application.

Recently (Doucet et al. 2000) introduced Rao-Blackwellized particle filters as an efficient solution to the SLAM problem. This approach has been followed by several authors, using a feature based approach (Montemerlo *et al.* 2002) with the name of FastSLAM (Factored Solution to SLAM) and also with grid mapping representations (Eliazar and Parr, 2003; Grisetti *et al.* 2005). The former allows a very compact representation of the map, but relies on extractors of predefined features. The latter can easily represent any object of the environment but requires large amounts of memory. The main drawback of the particle filter approach is its computational complexity measured in the number of required particles to build a map. The approaches (Haehnel et al. 2003; Montemerlo et al. 2003) try to reduce that number with more accurate sampling distributions.

Real use of the FastSLAM algorithm has been done only in outdoor environments, and real experiments have been done just with the Victoria Park data set, using point features. Successful mapping of that environment with just one particle let us think that an incremental maximum likelihood mapping is probably also able to successfully build that map, with the desired topological correctness. Unfortunately, this is not the case for a typical indoor environment, and some problems related to particle filters arise while implementing a feature based approach to the FastSLAM indoor application.

This paper presents a novel factorization of the SLAM problem that has interesting potential advantages. This factorization has been implemented for application to indoor environments, letting us to, up to our knowledge, successfully build for the first time indoor feature based maps with particle filters.

Section 2 presents an overview of mapping with particle filters, focused on feature based formulations, introducing the SLAM problem and the notation used and it describes the motivation of our novel formulation, presented in section 3. Implementation details are given in section 4. Experiments are presented in section 5 and conclusions in section 6.

2. FASTSLAM

With any feature based mapping approach, the environment is supposed to be composed by 'n' distinguishable landmarks, and each landmark $\Theta = \theta_1, \ldots, \theta_n$ is typically characterized by its location in space. The robot pose at timestep 't' will be denoted as s_t. Then the SLAM problem can be expressed in probabilistic terms as the posterior of map and robot pose given the history of all available measurements: z^t is the sequence of observations $z_1, \ldots, z_t$, and u^t is the sequence of controls $u_1, \ldots, u_t$

$$p(s_t, \Theta_t \mid z^t, u^t) \tag{1}$$

The robot pose evolves according to a probabilistic law, known as the motion model, which is a probabilistic function of the previous pose and the respective control. The robot sensor measurements also follow a probabilistic law, the observation model. This model is a function of the robot pose, the map, and the data association at time step 't', represented by variable 'n_t', that identifies which map feature correspond to the current observation.

$$\begin{aligned} &p(z_t \mid s_t, \Theta_t) = g(s_t, \theta_{n_t}) + \varepsilon_t \quad \text{(Observation)} \\ &p(s_t \mid s_{t-1}, u_t) = h(s_{t-1}, u_t) + \delta_t \quad \text{(Motion)} \end{aligned} \tag{2}$$

The motion and observation models have independent white noise with covariance matrices Q_t and R_t respectively.

Regular FastSLAM factorizes the posterior (1) as:

$$\begin{aligned} p(s^t, \Theta_t \mid z^t, u^t) &= p(s^t \mid z^t, u^t)\, p(\Theta_t \mid s^t, z^t, u^t) \\ &= p(s^t \mid z^t, u^t) \prod_{i=1}^{n} p(\theta_i \mid s^t, z^t, u^t) \end{aligned} \tag{3}$$

FastSLAM uses a particle filter to represent the first term, in which each particle represents an hypothesis about a robot path. Each path has attached 'n' independent Kalman filters to estimate the feature locations. There are two versions of the algorithm. Version 1.0 uses the following sampling distribution:

$$s_t^{[m]} \sim p(s_t \mid u_t, s_{t-1}^{[m]}) \tag{4}$$

This sampling distribution is the motion model of the robot. Typical large odometry uncertainty and high sensor accuracy (as it is the case of laser scanners), easily produce in this version what is known as the particle depletion problem. To reduce its effect (Montemerlo et al. 2003) propose the improved sampling distribution, which is basically the same idea described in (Grisetti et al. 2005) for grid maps

$$s_t^{[m]} \sim p(s_t \mid u^t, z^t, s_{t-1}^{[m]}) \tag{5}$$

Under this sampling distribution, the last measurement is integrated considering the history of past measurements, drastically reducing the meaningful space of the sampled variable.

Once the robot path is sampled, the map attached to each particle is updated, following EKF-style equations (Montemerlo et al. 2003) or with standard grid mapping techniques. This per particle map update can be typically done in constant time.

If the previous particle set is distributed following $p(s^{t-1} \mid z^{t-1}, u^{t-1})$, which is an asymptotically correct assumption, the resulting set of particles will be distributed according to the product between this function and the respective sampling distribution. This product is known as the proposal distribution. To account for the mismatch between the proposal distribution and the target one the importance weight of each particle is computed as:

$$w_t^{[m]} = \frac{\text{target}}{\text{proposal}} = \frac{p(s^t \mid z^t, u^t)}{p(s^{t-1} \mid z^{t-1}, u^{t-1})(\text{sampling})} \tag{6}$$

In the resampling step, particles with a low importance weight are replaced by samples with a high weight, to account for the use of a finite set of particles and to adjust the particle set to the target distribution.

The computational cost of the algorithm is dominated by the particle copying required in this resampling step, with a theoretical complexity of O(Kn), being K the number of particles. Nevertheless this complexity can be reduced to O(Klogn) by an adequate tree representation and management of the map features.

The problem of FastSLAM inconsistency has been recently reported by (Bailey et al. 2006). Our implementation of FastSLAM showed high difficulty in closing even small loops. It was noted that the very low uncertainty of the observations, derived from the feature extractor (i.e. a wall can be detected and extracted with a very high accuracy) was causing a particle depletion problem, requiring a prohibitive number of particles. The sampling step (4) produced many particles with a very low weight at each time step. Also, sampling with (5) reduces the problem while the robot is moving in a known area, and the measurements are integrated in the sampling distribution, reducing the sampling space. Unfortunately, if the observations do not correspond to any known feature (i.e. new feature, robot exploring new areas), the sampling distribution remains quite large, due to the typically large odometry uncertainty.

The improved proposal distributions of (Montemerlo et al. 2003), (Grisetti et al. 2005) are actually performing a kind of localization in the given map, to sample afterwards from the resulting robot pose conditioned probability function. It is important to note that the grid map based approaches do not retain any uncertainty in the map attached to each particle. The resulting behavior is like localizing the robot in the map, and then sampling from the possible maps. The formalization of this idea is what motivates our formulation presented in the following section.

3. DUAL FASTSLAM

In this section, a novel formulation to the SLAM problem is presented. This formulation is the dual of the FastSLAM (Montemerlo et al. 2002) factorization of (1), with an improved sampling distribution as in (Montemerlo et al. 2003). Dual FastSLAM is based on the other possible factorization of the posterior, in which data association variables have been omitted for clarity:

$$p(s_{t,}\Theta_t \mid z^t, u^t) = p(\Theta_t \mid z^t, u^t)\, p(s_t \mid \Theta_t, z^t, u^t) \quad (7)$$

The first factor represents the probability of a map given all the measurements, and the second term represents the probability of having a robot pose given a known map and all the data. Put verbally, if the map was known, the problem would be reduced to a standard robot localization problem.

Dual FastSLAM uses a particle filter of K particles to estimate (7). Each particle $\Theta_t^{[m]}$ contains an hypothesis of a possible map composed by ‘n_m‘ map features $\theta_{t,i}^{[m]}$, and it has attached a Gaussian of mean $\hat{s}_t^{[m]}$ and covariance matrix $P_t^{[m]}$.to represent the robot pose conditioned to that map:

$$\begin{aligned} &\boldsymbol{\Theta}_t \triangleq \Theta_t^{[1]}, \ldots, \Theta_t^{[K]} \\ &\Theta_t^{[m]} \triangleq N(\hat{s}_t^{[m]}, P_t^{[m]}), \theta_{t,1}^{[m]}, \theta_{t,2}^{[m]}, \ldots., \theta_{t,n_m}^{[m]} \end{aligned} \quad (8)$$

Feature locations do not contain any uncertainty information, as long as the different particles map hypothesis account for this uncertainty. Thus, the map of each particle is handled as perfectly known.

A Sampling the map

The particle set $\boldsymbol{\Theta}_t$ is calculated incrementally from the previous $\boldsymbol{\Theta}_{t-1}$ and the measurements z_t, u_t sampling from the distribution:

$$\Theta_t^{[m]} \sim p(\Theta_t \mid \Theta_{t-1}^{[m]}, z^t, u^t) \quad (9)$$

Note that this distribution is equivalent to the improved one (5), in the sense that it also contains the information of the last observation. If (9) is conditioned to s_t with the law of total probability:

$$\begin{aligned} \Theta_t^{[m]} \sim \int & p(\Theta_t \mid \Theta_{t-1}^{[m]}, z^t, u^t, s_t) \\ & \times p(s_t \mid \Theta_{t-1}^{[m]}, z^t, u^t) ds_t \end{aligned} \quad (10)$$

The Markov assumption simplifies (10) to:

$$\Theta_t^{[m]} \sim \int p(\Theta_t \mid \Theta_{t-1}^{[m]}, z_t, s_t)\, p(s_t \mid \Theta_{t-1}^{[m]}, z^t, u^t) ds_t \quad (11)$$

This equation can be solved in closed form under the linear Gaussian assumption, but it has to be carefully handled as this assumption produces two different solutions depending on the data association.

Consider the second term of (11) before the integration of measurement z_t and after the control u_t, that is the predicted robot state for particle ‘m’ noted with the notation (t|t-1), following the EKF style:

$$\begin{aligned} & p(s_t \mid \Theta_{t-1}^{[m]}, z^{t-1}, u^t) = N(\hat{s}_{t|t-1}^{[m]}, P_{t|t-1}^{[m]}) \\ & \hat{s}_{t|t-1}^{[m]} = h(\hat{s}_{t-1}^{[m]}, u_t) \\ & P_{t|t-1}^{[m]} = H_s P_{t-1}^{[m]} H_s^T + Q_t \\ & H_s = \nabla_{s_{t-1}} h(s_{t-1,} u_t)\Big|_{s_{t-1} = \hat{s}_{t-1}^{[m]}} \end{aligned} \quad (12)$$

When measurement z_t arrives, the resulting innovation $v_{t,n_t}^{[m]}$ and its covariance $V_{t,n_t}^{[m]}$ for each particle ‘m’ can be computed for every feature n_t of the map of that particle as:

$$\begin{aligned} & v_{t,n_t}^{[m]} = z_t - g(\hat{s}_{t|t-1}^{[m]}, \theta_{t-1,n_t}^{[m]}) \\ & V_{t,n_t}^{[m]} = G_s P_{t|t-1}^{[m]} G_s^T + R_t^{[m]} \\ & G_s = \nabla_{s_t} g(s_{t,} \theta_{n_t})\Big|_{s_t = \hat{s}_{t|t-1}^{[m]}} \end{aligned} \quad (13)$$

Then, the Mahalanobis test is used to associate the measurement to the most likely map feature.

$$\begin{aligned} & d_{t,n_t}^{[m]} = \left(v_{t,n_t}^{[m]}\right)^T \left(V_{t,n_t}^{[m]}\right)^{-1} v_{t,n_t}^{[m]} \\ & \hat{n}_t = \arg\min d_{t,n_t}^{[m]} \end{aligned} \quad (14)$$

It is possible that the measurement doesn't correspond to any map feature (with a confidence level α) and cannot be used to correct the estimation. In this case, the robot pose estimation after the measurement integrations remains equal to the one obtained in (12). Formally:

$$if\,(d_{t,\hat{n}_t}^{[m]} > \mathcal{X}^2_{\dim(v_{t,\hat{n}_t}^{[m]}),\alpha})$$

$$\begin{aligned} p(s_t \mid \Theta_{t-1}^{[m]}, z^t, u^t) &= p(s_t \mid \Theta_{t-1}^{[m]}, z^{t-1}, u^t) \\ &= N(\hat{s}_{t|t-1}^{[m]}, P_{t|t-1}^{[m]}) \end{aligned} \quad (15)$$

If the inequality in (15) is not satisfied, then the robot state can be updated as:

$$\begin{aligned} & p(s_t \mid \Theta_{t-1}^{[m]}, z^t, u^t) = N(\hat{s}_t^{[m]}, P_t^{[m]}) \\ & K^{[m]} = P_{t|t-1}^{[m]} G_s^T \left(V_{t,\hat{n}_t}^{[m]}\right)^{-1} \\ & \hat{s}_t^{[m]} = \hat{s}_{t|t-1}^{[m]} + K v_{t,\hat{n}_t}^{[m]} \\ & P_t^{[m]} = (I - K^{[m]} G_s) P_{t|t-1}^{[m]} \end{aligned} \quad (16)$$

In this case, the first term in (11) is independent of the measurement and the robot state, and under the Markov assumption it degenerates to:

$$p(\Theta_t \mid \Theta_{t-1}^{[m]}, \cancel{z}_t, \cancel{s}_t) = \begin{cases} \infty & \text{for } \Theta_t = \Theta_{t-1}^{[m]} \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

Sampling from the resulting distribution leads to (18), that means that it is not really necessary to sample, because the map hypothesis remains unchanged:

$$\Theta_t^{[m]} = \Theta_{t-1}^{[m]} \quad (18)$$

On the other hand if inequality (15) holds, it means that a new feature has to be added to the map, and the sampling distribution (11) can be written under the Gaussian assumption as:

$$\begin{aligned} & \Theta_t^{[m]} = \begin{pmatrix} \Theta_{t-1}^{[m]} & \theta_{t,n_m+1}^{[m]} \end{pmatrix} \\ & \theta_{t,n_m+1}^{[m]} \sim N(f(\hat{s}_{t|t-1}^{[m]}, z_t), F_s P_{t|t-1}^{[m]} F_s^T + R_t^{[m]}) \\ & \text{with} f \text{ from } \hat{z}_t = g(s_{t,} \theta_{n_t}) \Rightarrow \theta_{n_t} = f(s_{t,} \hat{z}_t) \\ & F_s = \nabla_{s_t} f(s_{t,} \hat{z}_t)\Big|_{s_t = \hat{s}_{t|t-1}^{[m]}, \hat{z}_t = z_t} \end{aligned} \quad (19)$$

That is, the previous map remains unchanged, and a new feature is added, which location is sampled from the given Gaussian.

3.1. Updating the robot pose

Once the map has been sampled, the robot pose has to be updated according to the new map, to represent the second term in (7). If the inequality in (15) is not satisfied, meaning that the measurement was associated to a previously known feature, then (18) implies that:

$$p(s_t \mid \Theta_t^{[m]}, z^t, u^t) = p(s_t \mid \Theta_{t-1}^{[m]}, z^t, u^t) \quad (20)$$

and the robot update done in (16) is valid. Otherwise, the robot pose estimation obtained in (15) has to be updated according to the new feature, performing a localization step as if the recently added feature was previously known.

$$\begin{aligned} & p(s_t \mid \Theta_t^{[m]}, z^t, u^t) = N(\hat{s}_t^{[m]}, P_t^{[m]}) \\ & K^{[m]} = P_{t|t-1}^{[m]} G_s^T \left(V_{t,n_m+1}^{[m]} \right)^{-1} \\ & \hat{s}_t^{[m]} = \hat{s}_{t|t-1}^{[m]} + K^{[m]} v_{t,n_m+1}^{[m]} \\ & P_t^{[m]} = (I - K^{[m]} G_s) P_{t|t-1}^{[m]} \end{aligned} \quad (21)$$

3.2. Importance weights

To account for the mismatch between the proposal distribution and the target one, the particles are resampled with likelihood proportional to their importance weights, computed as follows:

$$w_t^{[m]} = \frac{p(\Theta_t^{[m]} \mid z^t, u^t)}{p(\Theta_t^{[m]} \mid \Theta_{t-1}^{[m]}, z^t, u^t) p(\Theta_{t-1}^{[m]} \mid z^{t-1}, u^{t-1})} \quad (22)$$

Using the definition of conditional probability in the numerator:

$$\begin{aligned} w_t^{[m]} &= \frac{p(\Theta_t^{[m]} \mid \Theta_{t-1}^{[m]}, z^t, u^t) p(\Theta_{t-1}^{[m]} \mid z^t, u^t)}{p(\Theta_t^{[m]} \mid \Theta_{t-1}^{[m]}, z^t, u^t) p(\Theta_{t-1}^{[m]} \mid z^{t-1}, u^{t-1})} \\ &= \frac{p(\Theta_{t-1}^{[m]} \mid z^t, u^t)}{p(\Theta_{t-1}^{[m]} \mid z^{t-1}, u^{t-1})} \end{aligned} \quad (23)$$

Expanding the numerator using bayes rule:

$$\begin{aligned} w_t^{[m]} &\propto \frac{p(z_t \mid \Theta_{t-1}^{[m]}, z^{t-1}, u^t) p(\Theta_{t-1}^{[m]} \mid z^{t-1}, u^t)}{p(\Theta_{t-1}^{[m]} \mid z^{t-1}, u^{t-1})} \quad (24) \\ &\propto p(z_t \mid \Theta_{t-1}^{[m]}, z^{t-1}, u^t) \end{aligned}$$

and integrating over s_t based on the law of total probability:

$$\begin{aligned} w_t^{[m]} &\propto \int p(z_t \mid \Theta_{t-1}^{[m]}, z^{t-1}, u^t, s_t) \\ & \qquad p(s_t \mid \Theta_{t-1}^{[m]}, z^{t-1}, u^t) ds_t \quad (25) \\ &\propto \int p(z_t \mid \Theta_{t-1}^{[m]}, s_t) p(s_t \mid \Theta_{t-1}^{[m]}, z^{t-1}, u^t) ds_t \end{aligned}$$

This integral can be solved in closed form under the linear Gaussian assumption:

$$w_t^{[m]} \propto \frac{1}{(2\pi)^{\frac{n}{2}} \left| V_{t,\hat{n}_t}^{[m]} \right|} \exp\left(-\frac{1}{2} d_{t,\hat{n}_t}^{[m]} \right) \quad (26)$$

After the computation of this importance weight that is done per particle, these weights are normalized so they sum up to one. Resampling is necessary, as only a finite set of particles is used, but it can delete good samples, causing particle depletion. This problem can be reduced using a selective resampling based on the criterion of the effective number of particles:

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} \left(w_t^{[m]} \right)^2} \quad (27)$$

Our approach is the same as (Grisetti et al. 2005), so the resampling step is carried out every time N_{eff} drops below the threshold of N/2

4. IMPLEMENTATION

4.1. Environment Model

The features present in an indoor environment have an important associated problem: they are typically partially observable. Due to occlusions, noise or limited sensor range, it is very common that only a part of a whole wall (segment) can be observed. Thus, observing a segment does not provide full information to correct the robot pose, the segments contain information for the EKF equations only in two dimensions: the angular and lateral displacement. We have adapted to our algorithm the formulation presented in (Rodriguez-Losada *et al.* 2006) that inherits from the SPMap (Castellanos *et al.* 1999) approach which is the best existing solution to this problem.

The main idea is that the state of every feature is decomposed in the estimation of its state and an error vector that accounts for the difference between this estimation and the actual state of the object. The state vector of every object contains a reference frame attached to it and some objects might require extra parameters for its representation (e.g. a segment is modeled by a reference frame in its midpoint and its length).

4.2. Multiple observations

In the previous formulation, it has been supposed that each measurement z_t is composed by the observation of just one feature of the environment. In reality, it is common that the robot can simultaneously observe several features at the same time without changing its position. This multiplicity can be conveniently used to improve the performance of the algorithm.

When observing multiple features, the first step is the single data association performed by the Mahalanobis distance test (14). Then, a joint compatibility test can be done to check for the most likely simultaneous correct data association, computing a joint innovation vector with all the single compatible pairings, a joint innovation covariance matrix, and the resulting joint Mahalanobis distance. If this test is not passed, it means that at least one pairing that passed the single compatibility test is not compatible with the rest. To efficiently search for the most likely compatible set of pairings, an incremental computation of the inverse of the joint covariance matrix is done, as explained in the Joint Compatibility Branch and Bound (Neira and Tardos, 2001)

The resulting set of compatible pairings is applied first, so the robot pose computed in (16) is improved before performing the necessary map sampling (19) with the unpaired observations.

4.3. Computational cost

The presented formulation has a computational complexity of O(Kn), because of two reasons: particle copying at the resampling step, and the required iteration over all features in the per particle data association step. The first source of computational requirements can be highly reduced if it is considered that the particles can share large parts of a map. Thus, an adequate tree representation of the map features can avoid the copy of the whole map every time a particle is copied.

The data association step requires linear time in the number of map features, because all of them are compared with the observation to check correspondence. If a lookup table (a grid that roughly divides the space) is built while adding new features of the map, and the density of objects is finite, then this step could be done in constant time.

Because this work is focused on the formulation and the performance of the dual factorization, these optimizations have not been implemented yet, being subject of further development.

5. EXPERIMENTS

The algorithm has been fully implemented in C++. Several experiments have been done with different data sets from some real operating service robots, because our final goal is to have a robust inexpensive SLAM algorithm that can run onboard in real time, even in low-end processors as the Guido one, the smart walker of Haptica who provides a mobility aid for people who are both frail and visually impaired.

Figure 2. Guido the smart walker (left), and Urbano the interactive tour guide robot (right)

Guido mapping algorithm is based on standard SLAM-EKF that runs on an external laptop (Rodriguez-Losada et al. 2006). Although its performance is fine for small-medium size environments, its complexity is still quadratic. Furthermore, it relies on strong assumptions on the orthogonal shape of the environment to apply shape constraints that minimize the linearization errors that produce inconsistency in the estimation. While this approach works fine in such orthogonal environments it will most likely fail in other less structured environments. The data set was obtained in St. Mary' Hospital for the Blind with an exploration of 170 meters in 3 minutes and 30 seconds, when the robot was installed for field trials purposes.

The built map has 150 features and the loop is successfully closed 100% of times (25 runs) using just 30 particles, despite the large odometric error represented in Fig 3. Both the topological correctness and the orthogonal shape of the environment can be appreciated, despite any shape constraints has been used.

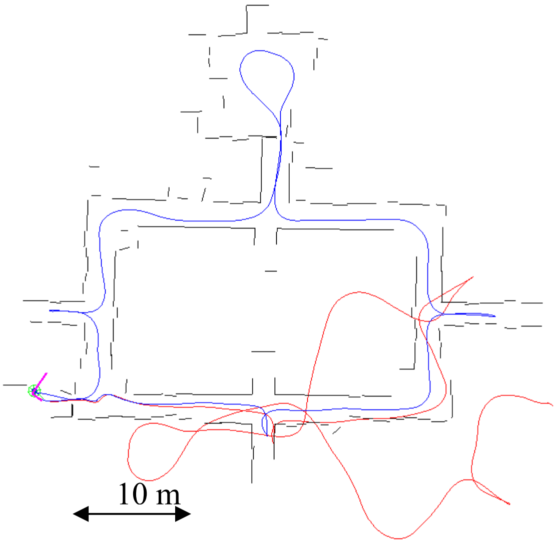

Figure 3: St. Mary's Hospital map

The second data set is provided by Urbano (Fig 2), a B21r tour guide robot, while exploring a populated trade fair. Urbano is a mobile interactive robot that is able to do guided visits, using the same mapping software as Guido. The exploration trajectory was done in 5 minutes, and covered 70 meters. This environment is more challenging as the density of features is lower and there are many occlusions due to people. One hundred particles were needed to successfully build this map (Fig 4). Orientation of segments is shown to see that close segments correspond to both sides of the walls, not to duplicated features.

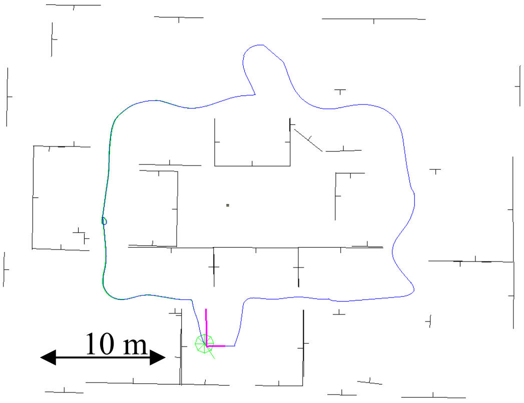

Figure 4. Indumatica 2005 trade fair map.

The last experiment was done with the MIT infinite corridor data set (just the first part). Because of the nested loops, the large trajectories and the large odometry error, three hundred particles were required to build the map of Fig. 5. Standard scan matching techniques were applied when features couldn't be extracted, to avoid a very large odometry uncertainty, applied once for the particle set, not per particle as (Grisetti et al. 2005) Despite the larger number of particles than in (Grisetti et al. 2005), and the lack of optimizations, it seems that the algorithm runs significantly faster. This point has to be confirmed, tough.

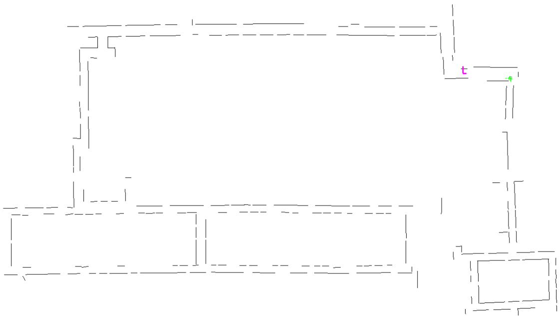

Figure 5. MIT infinite corridor (first part)

6. CONCLUSIONS

This paper has presented a novel dual factorization of the SLAM problem probabilistic formulation that has been successfully applied to indoor environments, building indoor feature based maps with particle filters for the first time.

Further research has to be done: It must be studied how to automatically select and dynamically adjust a number of particles, and how the different parameters, specially the odometry and sensor noise affect the performance of the algorithm. We also consider implementing some of the existing techniques to recover particle diversity after closing loops, which is very useful for mapping nested loops. The optimizations presented above have to be implemented to test the real time performance of the algorithm and its application to online map building.

We are also working in more extensive evaluation of the algorithm with comparison against FastSLAM and FastSLAM2.0.

7. ACKNOWLEDGMENT

We wish to thank Dr. G. Lacey and Haptica for Guido data set, and Dr. J. Leonard for MIT data set, and Dr. T. Bailey for the FastSLAM matlab source code in his webpage.

8. REFERENCES

Bailey T., Nieto J.and Nebot E. (2006) "Consistency of the FastSLAM Algorithm." IEEE International Conference on Robotics and Automation, 2006. Orlando, FL, USA. Page(s):424 – 429.

Castellanos J.A, Montiel J.M, Neira J, Tardos J.D. (1999) "The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building" *IEEE Trans. on Rob.& Aut.,* v. 15, n5.

Castellanos J.A., Neira J., Tardos J.D., (2004). "Limits to the consistency of EKF-based SLAM". Fifth IFAC Symposium on Intelligent Autonomous Vehicles IAV'04. Lisbon, Portugal.

Dissanayake G., Newman P., Clark S., Durrant-Whyte H., Csorba M., (2001). "A solution to the simultaneous localization and map building (SLAM) problem". IEEE Trans. on Robot& Aut.. v17.n.3 pp.229-241

Doucet A., Freitas J.F.G. de, Murphy K., Russel S. (2000) Raoblackwellized particle filtering for dynamic bayesian networks. Conf. on Uncertainty in Artifcial Intelligence (UAI).

Eliazar A. and Parr R. (2003) DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI).

Guivant J, and E. Nebot, (2001) "Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation" IEEE Trans. Rob.& Aut., v.17 n 3, p242-257

Grisetti G., Stachniss C., and Burgard W. (2005) Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. IEEE ICRA'05 pp 2443-2448, Barcelona, Spain

Hahnel D, Burgard W, Fox D., Thrun S. (2003) An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. IEEE IROS03.

Julier S., Uhlmann J.K., (2001). "A counter example to the theory of simultaneous localization and map building". IEEE ICRA´01. Vol. 4. Seul, Corea. pp. 4238-4243.

Montemerlo M., Thrun S., Koller D., Wegbreit B., (2002). "FastSLAM: A factored solution to the simultaneous localization and mapping problem". AAAI Nat.Conf. on Artif. Intelligence. Edmonton, Canada.

Montemerlo M., Thrun S., Koller D., Wegbreit B. (2003): FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. IJCAI. 1151-1156

Neira J. and Tardos J.D., (2001) "Data association in stochastic mapping using the joint compatibility test" *IEEE Trans. on Robotics and Automation,* vol. 17, no. 6, pp. 890-897.

Rodriguez-Losada D., Matia F., Galan R. (2006) Building geometric feature based maps for indoor service robots. *Elsevier: Robotics and Autonomous Systems.* v. 54, Issue 7 pp 546-558

Rodriguez-Losada D., Matia F., A. Jimenez, Galan R. (2006) Local map fusion for real-time indoor simultaneous localization and mapping. *Journal of Field Robotics.* Vol 23, Issue 5, p 291-309

Smith R., Self M., Cheeseman P. (1988) "Estimating uncertain spatial relationships in robotics" Uncertainty in Artificial Intelligence 2, J.F. Lemmer, L. N. Kanal, Eds. New York: Elsevier

Thrun. S. (2002). "Robotic mapping: A survey" In G. Lakemeyer and B. Nebel, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann.

Williams S., Dissanayake G., and Durrant-Whyte H.F. (2002) An efficient approach to the Simultaneous Localization and Mapping Problem ICRA'02 Washington DC, 15(5). pp 406-411