amcs

SIMULTANEOUS LOCALIZATION AND MAPPING: A FEATURE–BASED PROBABILISTIC APPROACH

PIOTR SKRZYPCZYŃSKI

Institute of Control and Information Engineering
Poznań University of Technology, ul. Piotrowo 3A, 60–965, Poznań, Poland
e-mail: `piotr.skrzypczynski@put.poznan.pl`

This article provides an introduction to Simultaneous Localization And Mapping (SLAM), with the focus on probabilistic SLAM utilizing a feature-based description of the environment. A probabilistic formulation of the SLAM problem is introduced, and a solution based on the Extended Kalman Filter (EKF-SLAM) is shown. Important issues of convergence, consistency, observability, data association and scaling in EKF-SLAM are discussed from both theoretical and practical points of view. Major extensions to the basic EKF-SLAM method and some recent advances in SLAM are also presented.

Keywords: mobile robot, navigation, simultaneous localization and mapping, feature matching.

1. Introduction

Determining the pose of a moving robot (self-localization) in the surrounding environment is one of the classic problems of mobile robotics. In a known environment, precise information about its pose enables the robot to plan further movements and activities (Skrzypczyński, 2005), while in an unknown environment, a pose estimate is required to build a model of this environment. Whenever a wheeled robot moves on a plane, the pose is defined as the position and orientation $\mathbf{x}_R = [x_r \ y_r \ \theta_r]^T$ with respect to an external coordinate system.

If the map is given *a priori* (i.e., it is predefined), the robot can self-localize by matching some features of the environment observed at the given moment to features of the same type existing in the known map. A feature suitable for self-localization is a static, salient object (or part of an object) in the environment, which can be described with respect to some co-ordinate frame. The acquisition of a precise, complete and up-to-date environment model for self-localization is a tedious task, particularly in a large environment, and it is hard to keep such a map up-to-date, as the environment may change over time. Moreover, a map surveyed by hand can be perceptually incompatible, i.e., it may not properly reflect all features in the environment perceived with the given sensing modality. Therefore, the robot should be able to build its own model of the environment.

This leads to the following "the chicken or the egg" dilemma: for self-localization, the robot requires a map, but to build such a map, the pose of the robot must be known. A solution to this problem is to build the map while computing a pose estimate, which is known as Simultaneous Localization And Mapping (SLAM).

In SLAM, the same sensory data are used to compute the robot pose and to construct the map. Hence, the robot pose estimate and the estimates of environment features are all correlated. Locations of discovered environment features in the map are uncertain, as well as the spatial relations between the features and the robot, because the underlying sensory data are uncertain. For these reasons, any mathematically rigorous solution to the SLAM problem must employ a framework that enables the representation, manipulation and propagation of the spatial uncertainty. Among several mathematical frameworks that fullfil these requirements, the probabilistic one is favorable, because it has strong theoretical foundations that provide a way of proving convergence properties and optimality of the developed solutions.

A break-through in the research on environment modelling for autonomous mobile robots was the formulation of the mapping and self-localization problems, which were formerly considered separately, as a single estimation problem. Smith *et al.* (1990) in their seminal paper developed the concept of a stochastic map—a probabilistic model of the spatial relations between the robot and the objects observed in its environment. They proposed to use

the Extended Kalman Filter (EKF) to estimate the state of the stochastic map. The increase in computational power available to autonomous robots and the introduction of commercial, affordable laser scanners made it possible to implement the stochastic map idea on real mobile robots (Castellanos and Tardós, 1999; Dissanayake *et al*., 2001).

Over the past decade, simultaneous localization and mapping has been one of the most dynamically developing fields in robotic research. Nowadays, the mainstream of SLAM research is focused on various improvements of the classic EKF-based SLAM solution (EKF-SLAM), while another trend is to use other, more recent probabilistic paradigms, like particle filtering (Arulampalam *et al*., 2002). Besides the probabilistic approach, some researchers adopted alternative uncertainty descriptions and data fusion techniques, e.g., set-theoretic estimators (DiMarco *et al*., 2004). Although the set-theoretic framework makes fewer assumptions as to the underlying uncertainty models (process disturbances and measurement errors are unknown but bounded), it suffers from other problems, related to the computation of set-valued estimates in high dimensional state spaces, typical for the SLAM problem. Similar issues are known for fuzzy-set-based localization and mapping methods (Gasós and Rosetti, 1999). There is no evidence that any alternative technique tried so far outperforms the probabilistic approach in practical SLAM scenarios.

In recent years, the research on SLAM applications has been heading towards more challenging environments: sub-sea, sub-terrain, unstructured, dynamic, etc., and towards the use of sensing modalities other than laser scanners, most notably passive vision sensors (cameras) (Paz *et al*., 2008).

This work deals with a probabilistic SLAM formulation and a feature-based environment representation, and it is focused on EKF-SLAM, which is the most frequently implemented SLAM framework, considered the reference algorithm to be compared with other methods. The use of the Kalman filtering ensures a fully analytical character of the solution and enables the use of established control theory methods to examine such properties of this solution as convergence, observability, and controllability.

The aim of this paper is to analyse the properties of EKF-SLAM cast as a state estimation problem, and to survey the major extensions of this method. Selected problems related to the implementation of EKF-SLAM in semi-structured indoor environments are also discussed.

2. Probabilistic approach to SLAM

2.1. Problem definition. In the probabilistic form of the SLAM problem, it is assumed that the system state, containing both the robot pose $\mathbf{x}_{R(k)}$ and the map $\mathbf{m}_{(k)}$, is represented at every time step k by some Probability Density Function (PDF). To solve the SLAM problem, it is necessary to find the posterior probability distribution at time k, assuming that all observations $\mathbf{z}^k = \{\mathbf{z}_{(1)}, \mathbf{z}_{(2)}, \ldots, \mathbf{z}_{(k)}\}$ and all control inputs $\mathbf{u}^k = \{\mathbf{u}_{(1)}, \mathbf{u}_{(2)}, \ldots, \mathbf{u}_{(k)}\}$ up to time k are known. The sought distribution, called the SLAM posterior, is defined as

$$p(\mathbf{x}_{R(k)}, \mathbf{m}_{(k)} \mid \mathbf{z}^k, \mathbf{u}^k). \tag{1}$$

The essential assumptions regarding the robot pose as well as the observations and the controls are the following:

- the robot states $\mathbf{x}_R$ follow a first-order Markov process:

$$p(\mathbf{x}_{R(k+1)} \mid \mathbf{x}_{R(1)}, \mathbf{x}_{R(2)}, \ldots, \mathbf{x}_{R(k)})$$
$$= p(\mathbf{x}_{R(k+1)} \mid \mathbf{x}_{R(k)}); \tag{2}$$

- errors in observations $\mathbf{z}$ are independent of errors in other observations;

- errors in controls $\mathbf{u}$ are independent of errors in other controls;

- errors in observations and controls at all time steps are uncorrelated.

To solve the above defined SLAM problem in its probabilistic form, the recursive Bayes filter can be employed. It extends the well-known Bayes rule to temporal estimation problems. Using that rule, the PDF (1) can be written in the following form:

$$p(\mathbf{x}_{R(k)}, \mathbf{m}_{(k)} \mid \mathbf{z}^k, \mathbf{u}^k)$$
$$= \eta p(\mathbf{z}_{(k)} \mid \mathbf{x}_{R(k)}, \mathbf{m}_{(k)}, \mathbf{z}^{k-1}, \mathbf{u}^k) \tag{3}$$
$$\times p(\mathbf{x}_{R(k)}, \mathbf{m}_{(k)} \mid \mathbf{z}^{k-1}, \mathbf{u}^k),$$

where η is a normalizing constant (Arulampalam *et al*., 2002), which ensures that (3) correctly represents a probability distribution. Taking into account that every observation $\mathbf{z}_i, i = 1, \ldots, k$ is statistically independent of the other observations and controls at the same time step, and then exploiting the Markov property (2) and using the total probability theorem, the PDF (3) can be rewritten in the form depending on the previous state of the robot and the map at time $k - 1$:

$$p(\mathbf{x}_{R(k)}, \mathbf{m}_{(k)} \mid \mathbf{z}^k, \mathbf{u}^k)$$
$$= \eta p(\mathbf{z}_{(k)} \mid \mathbf{x}_{R(k)}, \mathbf{m}_{(k)})$$
$$\times \iint p(\mathbf{x}_{R(k)}, \mathbf{m}_{(k)} \mid \mathbf{x}_{R(k-1)}, \mathbf{m}_{(k-1)}, \mathbf{z}^{k-1}, \mathbf{u}^k)$$
$$\times p(\mathbf{x}_{R(k-1)}, \mathbf{m}_{(k-1)} \mid \mathbf{z}^{k-1}, \mathbf{u}^k) \, \mathrm{d}\mathbf{x}_{R(k-1)} \, \mathrm{d}\mathbf{m}_{(k-1)}. \tag{4}$$

Equation (4) describes the most general SLAM version, where the map can change over time. However, using this equation to compute the SLAM posterior is very

expensive, because an integration over the high dimensional space of all maps $\mathbf{m}$ is required. Moreover, using (4) in practice requires a model of the map evolution over time, which, contrary to the robot motion model, is usually unavailable. Hence, a static world assumption is common in SLAM research. Under this assumption the robot is the only moving entity and the map $\mathbf{m}$ does not evolve over time. Hence the sought PDF takes the form $p(\mathbf{x}_{R(k)}, \mathbf{m} \mid \mathbf{z}^k, \mathbf{u}^k)$, and Eqn. (4) may be simplified to

$$
\begin{aligned}
& p(\mathbf{x}_{R(k)}, \mathbf{m} \mid \mathbf{z}^k, \mathbf{u}^k) \\
& = \eta p(\mathbf{z}_k \mid \mathbf{x}_{R(k)}, \mathbf{m}) \\
& \quad \times \int p(\mathbf{x}_{R(k)}, \mathbf{m} \mid \mathbf{x}_{R(k-1)}, \mathbf{z}^{k-1}, \mathbf{u}^k) \\
& \quad \times p(\mathbf{x}_{R(k-1)} \mid \mathbf{z}^{k-1}, \mathbf{u}^k) \, \mathrm{d}\mathbf{x}_{R(k-1)}.
\end{aligned}
\tag{5}
$$

Using the definition of conditional probability and again exploiting the Markov property, Eqn. (5) can be transformed to

$$
\begin{aligned}
& p(\mathbf{x}_{R(k)}, \mathbf{m} \mid \mathbf{z}^k, \mathbf{u}^k) \\
& = \eta \overbrace{p(\mathbf{z}_{(k)} \mid \mathbf{x}_{R(k)}, \mathbf{m})}^{\text{observation model}} \\
& \quad \times \int \overbrace{p(\mathbf{x}_{R(k)} \mid \mathbf{x}_{R(k-1)}, \mathbf{u}_{(k)})}^{\text{process model}} \\
& \quad \times \overbrace{p(\mathbf{x}_{R(k-1)}, \mathbf{m} \mid \mathbf{z}^{k-1}, \mathbf{u}^{k-1})}^{\text{previous state}} \, \mathrm{d}\mathbf{x}_{R(k-1)},
\end{aligned}
\tag{6}
$$

which is a recursive version of the Bayes filter for the SLAM problem. Because the formula (6) defines a probability distribution over a continuous space, it can be implemented only if additional assumptions are made as to the representation of the SLAM posterior as well as specification of the process dynamics model and the observation model.

2.2. Basic EKF-SLAM algorithm. The EKF-SLAM algorithm is one of the possible implementations of the general, probabilistic SLAM solution given by (6). It is based on the Kalman filter algorithm, which is a form of the Bayes filter that represents PDFs with Gaussian distributions.

To use the Kalman filter to estimate the SLAM posterior (1), two assumptions have to be met: (i) the robot motion model and the observation model must be linear with additive Gaussian noise, (ii) the SLAM posterior PDF must be defined as a Gaussian:

$$
p(\mathbf{x}_{R(k)}, \mathbf{m} \mid \mathbf{z}^k, \mathbf{u}^k) \sim N(\hat{\mathbf{x}}_{(k)}, \mathbf{C}_{x(k)}),
\tag{7}
$$

where the $\hat{\mathbf{x}}_{(k)}$ vector represents the system state at time k, and $\mathbf{C}_{x(k)}$ is the covariance matrix of this vector. The state vector contains the estimated pose of the robot $\hat{\mathbf{x}}_R$ and

the estimated locations of n features $\hat{\mathbf{x}}_{F_i}$, $(i = 1, \ldots, n)$. The spatial uncertainty of the estimated features, the robot pose, and their correlations are represented by the covariance matrix

$$
\hat{\mathbf{x}}_{(k)} = \begin{bmatrix} \hat{\mathbf{x}}_{R(k)} \\ \hat{\mathbf{x}}_{F_1(k)} \\ \hat{\mathbf{x}}_{F_2(k)} \\ \vdots \\ \hat{\mathbf{x}}_{F_n(k)} \end{bmatrix},
\tag{8}
$$

$$
\mathbf{C}_{x(k)} = \begin{bmatrix} \mathbf{C}_R & \mathbf{C}_{RF_1} & \ldots & \mathbf{C}_{RF_n} \\ \mathbf{C}_{F_1R} & \mathbf{C}_{F_1} & \ldots & \mathbf{C}_{F_1F_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{F_nR} & \mathbf{C}_{F_nF_1} & \ldots & \mathbf{C}_{F_n} \end{bmatrix}.
\tag{9}
$$

The Kalman filter provides an optimal least squares error estimate of the SLAM posterior given the information at hand, assuming that the system model is linear. However, all real SLAM systems are non-linear, hence the linearised EKF must be employed.

In wheeled mobile robots the displacement measurement provided by encoder-based odometry is used as the control input $\mathbf{u}$ at time k, because it usually provides a better estimate of the vehicle movement than just a mathematical motion model. To make the general SLAM state model independent of the source of proprioceptive data (e.g., inertial sensors and rate gyros can be used here instead of wheel encoders), odometry is usually considered an external sensing system, which has its own mathematical model. However, in some systems relying on exteroceptive sensing only (e.g., vision), the state vector is extended by an ego-motion estimate, which is considered jointly with the robot pose and environment features (Davison *et al.*, 2007). Such an approach may be particularly beneficial for walking robots and vehicles using the "skid-steering" principle, which causes high slippage and makes position tracking extremely difficult (Kozłowski and Pazderski, 2004).

Although the assumption that a robot pose has the Markov property coincides with an intuition that to reach a given state (pose) the robot had to perform a known action (movement) from the previous state, in some situations, when the movement of the robot is not a result of any known action, the Markov property does not hold. Such a situation is termed "teleportation" or a "kidnap situation" in the literature. From the point of view of the Kalman filter formalism, this is a sudden state change that is not captured by the process model—in such a case the model ceases to follow the behaviour of the system, so it is obvious that the filter cannot handle such a situation without an external recovery mechanism.

The observation model is determined by the procedure employed to predict observations on the basis of the current robot state and information from the map. This

model is described by a non-linear function $\mathbf{h}$. The assumption that the observations made at two different time steps are independent usually holds whenever these observations are taken from different vantage points. In many mobile robots, navigation is accomplished in a stop-and-go manner, but whenever in-motion navigation is considered, temporal relations of observations from all the proprioceptive and exteroceptive sensors have to be maintained explicitly by proper time stamping (Arras *et al.*, 2001).

The procedure, given by (6), which updates the SLAM posterior can be divided into two stages: the prediction stage, which predicts the PDF at the next time step on the basis of the previous state and the process dynamics model, and the correction stage, which corrects the predicted PDF using the exteroceptive measurements (observations). Also the EKF algorithm can be divided into similar two stages (prediction and correction). If the previous PDF parameters $(\hat{\mathbf{x}}_{(k-1)}, \mathbf{C}_{x(k-1)})$ and the robot displacement $\mathbf{u}_{(k)}$ from $k-1$ to k are known, the updated PDF parameters at the k-th step are computed by applying the EKF prediction stage equations

$$\hat{\mathbf{x}}_{R(k|k-1)} = \mathbf{f}(\hat{\mathbf{x}}_{R(k-1|k-1)}, \mathbf{u}_{(k)}), \qquad (10)$$

$$\hat{\mathbf{x}}_{F_i(k+1)} = \hat{\mathbf{x}}_{F_i(k)}, \qquad (11)$$

$$\mathbf{C}_{R(k|k-1)} = \mathbf{J}_{f_R} \mathbf{C}_{R(k-1|k-1)} \mathbf{J}_{f_R}^T$$
$$+ \mathbf{J}_{f_U} \mathbf{C}_{u(k)} \mathbf{J}_{f_U}^T, \qquad (12)$$

$$\mathbf{C}_{RF_i(k|k-1)} = \mathbf{J}_{f_R} \mathbf{C}_{RF_i(k-1|k-1)}, \qquad (13)$$

where $\mathbf{C}_u$ is the displacement vector covariance matrix, $\mathbf{f}$ is a non-linear function describing how the state evolves with the given displacement, while $\mathbf{J}_{f_R}$ and $\mathbf{J}_{f_U}$ are the Jacobians of $\mathbf{f}$ with respect to $\mathbf{x}_R$ and $\mathbf{u}$, respectively. Notice that the part of the state vector related to the robot pose $\mathbf{x}_R$ is updated, cf., (10), but that related to the map remains unchanged, cf., (11), as, according to the static world assumption, it has no dynamics. In the prediction stage, the robot pose uncertainty grows, because the pose is changed without integrating any new information from an independent source. This stage has a linear computational complexity with respect to the number of map features.

In the SLAM framework, the uncertainty in both the robot pose and the feature locations is decreased by re-observing the features. This implements the correction stage of (6). At first, observations of features are predicted using the map and the robot pose updated by the previous EKF stage:

$$\hat{\mathbf{z}}_{(k)} = \begin{bmatrix} \hat{\mathbf{z}}_{1(k)} \\ \hat{\mathbf{z}}_{2(k)} \\ \vdots \\ \hat{\mathbf{z}}_{n(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{h}(\hat{\mathbf{x}}_R, \mathbf{x}_{F_1}) \\ \mathbf{h}(\hat{\mathbf{x}}_R, \mathbf{x}_{F_2}) \\ \vdots \\ \mathbf{h}(\hat{\mathbf{x}}_R, \mathbf{x}_{F_n}) \end{bmatrix}, \qquad (14)$$

$$\mathbf{C}_{z(k)} = \mathbf{J}_{h_X} \mathbf{C}_{x(k|k-1)} \mathbf{J}_{h_X}^T, \qquad (15)$$

where $\mathbf{h}$ is a non-linear measurement function that transforms the features from the global map co-ordinates into the local robot frame, while $\mathbf{J}_{h_X}$ is a Jacobian of $\mathbf{h}$ with respect to the state. Actual features observed at time k are described by the parameter vectors $\mathbf{z}_{j(k)}$ and the covariance matrices $\mathbf{C}_{r_j(k)}$, where $j = 1, \ldots, n_f$ is an observation index.

The general probabilistic SLAM solution (6) assumes implicitly that the pairings between the map features and the actual observations are known, i.e., the data association problem is already solved. In any real EKF-SLAM implementation, this problem has to be solved on-line, for every incoming observation $\mathbf{z}_{(k)}$. The standard feature matching technique is the squared Mahalanobis distance test. The so-called validation gate is defined around the predicted observation of the i-th feature:

$$(\mathbf{z}_{(k)} - \hat{\mathbf{z}}_{i(k)})(\mathbf{C}_{z_i(k)} + \mathbf{C}_{r(k)})^{-1}(\mathbf{z}_{(k)} - \hat{\mathbf{z}}_{i(k)})^T$$
$$\leq \chi^2_{r,\alpha}, \quad (16)$$

where the threshold value $\chi^2_{r,\alpha}$ is chosen from the χ^2 distribution with r degrees of freedom, while α is the significance level on which the hypothesis of pairing correctness is rejected. The new observation $\mathbf{z}_{(k)}$ is considered as matching the prediction if it fits into the validation gate. Because EKF-SLAM implements the pose prediction step, no global data association problem is assumed to exist. Thus the pairing candidate for an observation can always be found in the validation gate around the prediction of this observation.

During the operation of a real SLAM system, a data association ambiguity can arise whenever more than one observation pass the Mahalanobis distance test (16) and there are several statistically feasible pairings. An additional test that exploits constraints imposed by the geometry of features may be employed, depending on the type of geometric features used (points, corners, line segments, etc.) and the sensory system of the robot. Such a test is usually used prior to (16). However, it does not guarantee the uniqueness of pairings. Because of this, a matching strategy has to be defined. The most widely used strategy, classic in tracking applications (Bar-Shalom *et al.*, 2001), is the nearest neighbor filter, according to which the accepted pairing is the one that has the smallest Mahalanobis distance computed in (16).

A comparison of paired actual and predicted observations enables us to correct the initial prediction of the system state by integrating new information from the exteroceptive sensors. The successfully paired features are used to update the state vector and its covariance by applying EKF correction stage equations:

$$\mathbf{K}_{(k)} = \mathbf{C}_{x(k|k-1)} \mathbf{J}_{h_X}^T (\mathbf{J}_{h_X} \mathbf{C}_{x(k|k-1)} \mathbf{J}_{h_X}^T + \mathbf{C}_{r(k)})^{-1}, \qquad (17)$$
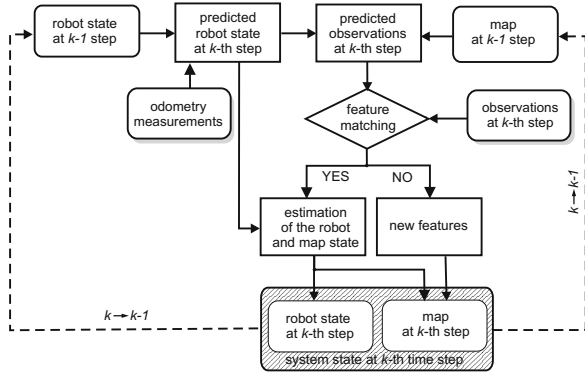
Fig. 1. Block diagram of the EKF-SLAM algorithm.

$$\hat{\mathbf{x}}_{(k|k)} = \hat{\mathbf{x}}_{(k|k-1)} + \mathbf{K}_{(k)}(\mathbf{z}_{(k)} - \hat{\mathbf{z}}_{(k)}), \qquad (18)$$

$$\mathbf{C}_{x(k|k)} = (\mathbf{I} - \mathbf{K}_{(k)}\mathbf{J}_{h_X})\mathbf{C}_{x(k|k-1)}. \qquad (19)$$

The computational complexity of the EKF-SLAM correction (estimation) stage is $O(n^2m)$, where m is the number of paired features.

Integrating the new information from an independent source allows reducing the uncertainty of the robot pose and the spatial uncertainty of map features. Due to the correlations in the system covariance matrix (9), new observations reduce the uncertainty of all features in the map, including those that cannot be observed from the current vantage point.

The environment features that have been observed but not paired with any feature from the map are added to the map, augmenting the state vector. The procedure of adding to the map a new feature upon the j-th observation is given by the equations

$$\hat{\mathbf{x}}_{F_{n+1}(k)} = \mathbf{g}(\hat{\mathbf{x}}_{(k|k)}, \mathbf{z}_{j(k)}), \qquad (20)$$

$$\mathbf{C}_{F_{n+1}(k)} = \mathbf{J}_{g_R}\mathbf{C}_{R(k|k)}\mathbf{J}_{g_R}^T + \mathbf{J}_{g_F}\mathbf{C}_{r_j(k)}\mathbf{J}_{g_F}^T, \quad (21)$$

$$\mathbf{C}_{F_{n+1}R(k)} = \mathbf{J}_{g_R}\mathbf{C}_{R(k)}, \qquad (22)$$

$$\mathbf{C}_{F_{n+1}F_i(k)} = \mathbf{J}_{g_R}\mathbf{C}_{RF_i(k)}, \qquad (23)$$

where $i = 1, \dots, n$, $\mathbf{g}$ is a non-linear function transforming the observed features from the robot co-ordinates into the global map co-ordinates, while $\mathbf{J}_{g_R}$ and $\mathbf{J}_{g_F}$ are the Jacobians of $\mathbf{g}$ with respect to the robot pose and the j-th observed feature parameters, respectively. The computational complexity at this stage is $O(nl)$, i.e., with respect to the number of map features n, while $l = (n_f - m)$ is the number of new features added to the map. The estimated state and its uncertainty at time k are treated as the initial estimate for the algorithm at the next time step $k+1$ (Fig. 1).

3. EKF-SLAM properties

3.1. Convergence of the EKF-SLAM algorithm.
Although SLAM has been a very active research area in mobile robotics for the last decade, very few analytical results on the essential properties of SLAM as an estimation problem have been published. The strongest results concerning the convergence of an SLAM algorithm are available for the Kalman filter based SLAM. Dissanayake *et al.* (2001) presented analytical proofs of the following properties concerning the convergence of SLAM and lower bound on the pose uncertainty.

Property 1. *The determinant of any submatrix of the map covariance matrix decreases monotonically with every observation made:*

$$\mathbf{C}_x = \begin{bmatrix} \mathbf{C}_R & \mathbf{C}_{RM} \\ \mathbf{C}_{RM}^T & \mathbf{C}_M \end{bmatrix},$$

$$\det\mathbf{C}_{M(k|k)} \le \det\mathbf{C}_{M(k-1|k-1)}. \qquad (24)$$

Property 2. *As the number of observations increases, the map feature estimates become more correlated, and fully correlated in the limit*

$$\lim_{k \to \infty}[\det\mathbf{C}_{M(k|k)}] = 0. \qquad (25)$$

Property 3. *The lower bound on the state vector covariance matrix $\mathbf{C}_x$, and thus also on the map accuracy (given by the sub-matrix $\mathbf{C}_M$) is a function of the initial robot pose uncertainty when the first feature was observed:*

$$\lim_{k \to \infty} \mathbf{C}_{x(k|k)}$$
$$= \begin{bmatrix} \mathbf{C}_{R_0} & \mathbf{C}_{R_0}\mathbf{H}_v^T\left(\mathbf{H}_{p_1}^T\right)^{-1} \\ \mathbf{H}_{p_1}^{-1}\mathbf{H}_v\mathbf{C}_{R_0} & \mathbf{H}_{p_1}^{-1}\mathbf{H}_v\mathbf{C}_{R_0}\mathbf{H}_{p_1}^{-1}\mathbf{H}_v^T \end{bmatrix}, \qquad (26)$$

where $\mathbf{C}_{R_0}$ is the initial robot pose uncertainty, while $\mathbf{H}_v$ and $\mathbf{H}_{p_1}$ are components of the observation matrix (Dissanayake *et al.*, 2001).

Although these three results fully define filter steady state performance, their proofs were conducted in (Dissanayake *et al.*, 2001) by examining a simple linear version of the SLAM problem. Thus, the properties hold only under the specific conditions: a linear motion model, a linear observation model, and an infinite number of feature observations. These conditions do not hold in any real SLAM implementation. For instance, due to the orientation term of the robot pose, all motion and observation models are non-linear in practice. However, these properties of the Kalman filter based SLAM algorithm have important implications:

- the uncertainty of any feature location in the map never increases;

- the correlation between any two features in the map never decreases;

- the uncertainty of any feature location in the map cannot decrease below its lower bound;

- the robot pose uncertainty cannot decrease below its initial value as a result of SLAM algorithm operation.

These rules can be used for simple, practical checking of EKF-SLAM consistency.

The convergence properties described in (Dissanayake *et al.*, 2001) were examined in (Huang and Dissanayake, 2006) for the case of non-linear process and observation models, which is more relevant to a practical EKF-SLAM implementation. Proofs of Properties 1 and 3 were conducted for a general, non-linear EKF-SLAM system. However, these proofs hold under the assumption that the Jacobians are evaluated at the true robot pose and the true feature locations, whereas in any real SLAM system these values are unknown, and the linearisation is performed around the current state estimate (prediction). Thus, the question whether the linearisation impacts convergence in real EKF-SLAM implementations remains open. Note also that all these theoretical results hold only assuming known (i.e., perfect) data associations.

3.2. Linearisation problem. Among the assumptions allowing the application of the EKF to solve the SLAM problem, the most commonly violated one concerns the use of linearised motion and observation models. EKF-SLAM linearises both the motion and observation models by using first-order Taylor series expansions around a working point, which is the current state estimate. The precision of the results obtained by the application of EKF is highly influenced by linearisation quality. EKF linearisation yields good results only if the modelled processes are approximately linear, and the working point is computed accurately, with low bias and uncertainty (Bar-Shalom *et al.*, 2001). In real EKF-SLAM applications, the functions describing both the robot motion and the feature observation contain considerable non-linearities, due to the orientation term of the robot pose. Moreover, the state estimate used as the working point is usually a prediction obtained from odometry, and thus it could be biased and highly uncertain.

It is well known that this kind of linearisation error can cause incorrect estimation and lead to EKF divergence (Bar-Shalom *et al.*, 2001). It is also reported in the literature (Castellanos *et al.*, 2004; Julier and Uhlmann, 2001) that EKF-SLAM may produce overconfident estimates of the robot pose and locations of the map features. These are symptoms of EKF-SLAM inconsistency, caused by linearisation errors in Jacobians and/or unmodelled residual process dynamics. Such modelling errors are known in the practice of general EKF applications, and the tuning (enlarging) of the process and observation covariance matrices in the filter or the injection of stabilizing noise are recommended as remedies (Maybeck, 1979). However, the specific properties of the SLAM problem make these general techniques ineffective.

The unmodelled errors in process dynamics are often non-stationary and temporally correlated, as in the case of an encoder-based odometry model, where the errors and even bias depend upon the local wheel-to-floor contact characteristics, and thus cannot be effectively calibrated or compensated in the filter (Skrzypczyński, 2007a). Moreover, the SLAM problem has a specific structure of the state space: a large part of it (the map) has no dynamics, thus stabilizing noise can be injected only into the robot state. Any artificial noise injected into the system may also impact the data association process.

It was shown in (Julier and Uhlmann, 2001) that EKF-SLAM becomes inconsistent due to Jacobian linearisation errors, with the suggestion that eventual inconsistency of the algorithm is inevitable after a longer time period. However, results of some recent theoretical and simulation works (Bailey *et al.*, 2006; Castellanos *et al.*, 2004; Huang and Dissanayake, 2006) show that filter inconsistency in SLAM is closely related to the robot orientation uncertainty. It has been shown using Monte Carlo simulations that keeping the true orientation uncertainty small enough prevents EKF-SLAM inconsistency for long time periods (Bailey *et al.*, 2006).

Similar conclusions were drawn in (Skrzypczyński, 2007a) from results of real-world experiments with an EKF-SLAM system implementation tested in an indoor environment. In EKF-SLAM implementations on wheeled mobile robots, the process modelling errors are caused mainly by unrealistic statistical models of odometry. A problem arises if not all aspects of the odometric uncertainty are correctly modelled by $\mathbf{C}_u$. However, such sources of errors like collisions or large bumps on the floor cannot be anticipated by any statistical odometry model. Therefore, one must recourse to other types of sensors to improve robot displacement prediction.

If proprioceptive sensors such as rate-gyros and compasses are not available, the displacement estimate can be obtained from an exteroceptive sensor. We showed in (Skrzypczyński, 2006) that the translational and the rotational displacement between two robot poses can be obtained by scan matching, i.e., by finding a rotation and translation that maximize the overlap of these scans. For the correction of odometry errors we match two consecutive scans to compute the incremental displacement. Integrating these displacements, given by the translation $\Delta\mathbf{t}_i = [\Delta x_i \ \Delta y_i]^T$ and the rotation $\Delta\theta_i$ for the pair of scans $i-1$ and i, we establish an alternative form of odometry, which yields more accurate pose estimates and offers a much more predictable uncertainty model (Skrzypczyński, 2007b). The correction of odometry errors enables regular updates of the orientation estimate between the consecutive time steps of EKF-SLAM (the
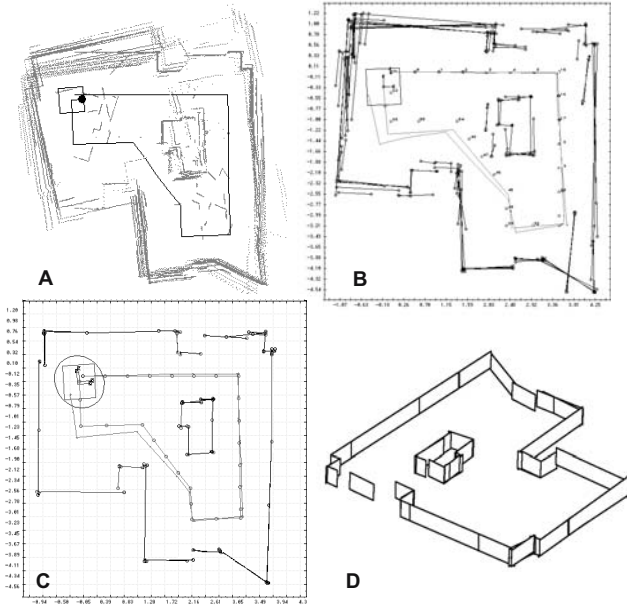
Fig. 2. Results for the experiment with unmodelled odometry errors (a), standard EKF-SLAM (b) and the proposed approach (c), (d).

orientation is updated more often than the features are observed), so the orientation uncertainty is kept small, and the SLAM algorithm produces consistent results.

An example of results obtained with this method of robot displacement estimation is shown in Fig. 2. This particular EKF-SLAM version, described in detail in (Skrzypczyński, 2007a), uses a robust geometric feature extraction method supported by a local grid-based environment representation, and has been implemented on a Labmate mobile robot equipped with a Sick LMS 200 laser scanner. Figure 2(a) shows the nominal robot path and raw laser scanner data registered in a global frame by using encoder-based odometry. It can be seen that the robot pose error grows without bounds, and the scans do not overlap. In this experiment, the parameters of the Labmate odometry model were roughly identified by an ad-hoc experiment, leaving the systematic errors uncalibrated. The first map (Fig. 2(b)) was built with the basic EKF-SLAM algorithm, and it is highly inconsistent. Clearly, state prediction and linearisation errors due to inaccurate odometry gradually prevented correct associations between the map and the segments from new observations. The segment-based map obtained from the scanner data gathered with the same experimental set-up but using scan matching to control the orientation uncertainty (Fig. 2(c)), is much more consistent. The estimates of robot displacement from scan matching prevented the EKF from large divergence in spite of the poor process model. Some segments are still multiplied, but the overall number of features in the second map is greatly reduced,

which is visible in the 3D view (Fig. 2(d)).

An interesting alternative approach to the non-linearity problem in Kalman filter based SLAM is the use of the Unscented Kalman Filter (UKF) instead of the classic EKF (Andrade-Cetto *et al.*, 2005). The UKF uses the non-linear process model directly, instead of linearising it with Jacobians. At each time step, the UKF samples the state around the current mean estimate, with a deterministic number of samples. Each sample is then propagated with the actual non-linear dynamics model, and a new mean is calculated. This new mean is considered to be the predicted state. It has been shown in simulations (Bailey *et al.*, 2006) that UKF-based SLAM degrades more slowly than EKF-SLAM under the same conditions. Although the introduction of the UKF appears to be insufficient to produce an SLAM algorithm completely immune to linearisation errors, UKF may be considered a useful extension, particularly for highly non-linear motion models.

3.3. Controllability and observability.

An interesting problem is the analysis of the controllability and observability of EKF-SLAM. An SLAM system can be described as a non-linear dynamical system, with its state and observation equations given as

$$\mathbf{x}_{(k+1)} = \mathbf{f}(\mathbf{x}_{(k)}, \mathbf{u}_{(k)}, \mathbf{v}_{(k)}),$$
$$\mathbf{z}_{(k+1)} = \mathbf{h}(\mathbf{x}_{(k)}) + \mathbf{w}_{(k)}, \tag{27}$$

where $\mathbf{v}_{(k)}$ and $\mathbf{w}_{(k)}$ are vectors of random, uncorrelated, zero-mean, Gaussian noise.

Although in SLAM the control signal $\mathbf{u}$ is replaced by proprioceptive measurements, the influence of process noise on the state vector is crucial to filter performance. Hence, we disregard the vector $\mathbf{u}$ henceforth, and investigate the controllability and observability with respect to the vector $\mathbf{v}$. The application of the Kalman filter to solve the SLAM problem requires the linearization of the system model and the observation model using the Taylor series expansion:

$$\mathbf{x}_{(k+1)} \approx \mathbf{x}_{(k+1|k)} + \mathbf{F}(\mathbf{x}_{(k)} - \mathbf{x}_{(k|k)}) + \mathbf{G}\mathbf{v}_{(k)},$$
$$\mathbf{z}_{(k+1)} \approx \mathbf{z}_{(k+1|k)} + \mathbf{H}(\mathbf{x}_{(k+1)} - \mathbf{x}_{(k+1|k)}) + \mathbf{w}_{(k+1)}, \tag{28}$$

where $\mathbf{F}$ and $\mathbf{G}$ are the Jacobians of the function $\mathbf{f}$ with respect to $\mathbf{x}$ and $\mathbf{v}$, respectively, while $\mathbf{H}$ is the Jacobian of the function $\mathbf{h}$ with respect to $\mathbf{x}$. To make the analysis of the influence of the disturbances on the state of the linearised system easier, Eqns. (28) are re-written, as proposed in (Andrade-Cetto and Sanfeliu, 2005), in terms of the state estimation error $\tilde{\mathbf{x}}_{(k|k)} = \mathbf{x}_{(k)} - \mathbf{x}_{(k|k)}$:

$$\tilde{\mathbf{x}}_{(k+1|k)} = \mathbf{F}\tilde{\mathbf{x}}_{(k|k)} + \mathbf{G}\mathbf{v}_{(k)},$$
$$\tilde{\mathbf{z}}_{(k+1|k)} = \mathbf{H}\tilde{\mathbf{x}}_{(k+1|k)} + \mathbf{w}_{(k+1)}. \tag{29}$$

The controllability matrix for the system given by (29) is

$$\mathbf{C} = \begin{bmatrix} \mathbf{G} & \mathbf{FG} & \mathbf{F}^2\mathbf{G} & \dots & \mathbf{F}^{\dim\mathbf{x}-1}\mathbf{G} \end{bmatrix}. \tag{30}$$

To facilitate the analysis of (30), we consider the specific structure of the SLAM state vector, which consists of $\mathbf{x}_{R(k)}$, the robot pose sub-vector at time k, and $\mathbf{x}_F$, the sub-vector containing n map features. Using this decomposition, and taking into account that the features are static and only the $\mathbf{x}_R$ part of the state $\mathbf{x}$ evolves, the first equation of (29) can be written in the Kalman controllable form

$$\begin{bmatrix} \tilde{\mathbf{x}}_{R(k+1|k)} \\ \tilde{\mathbf{x}}_{F(k+1|k)} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_{R(k|k)} \\ \tilde{\mathbf{x}}_{F(k|k)} \end{bmatrix} \\ + \begin{bmatrix} \mathbf{G}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{R(k)} \\ \mathbf{0} \end{bmatrix}, \quad (31)$$

where $\mathbf{F}_R$ and $\mathbf{G}_R$ are Jacobian matrices. The controllability matrix for the system given by (31) is

$$\mathbf{C} = \begin{bmatrix} \mathbf{G}_R & \mathbf{F}_R \mathbf{G}_R & \mathbf{F}_R^2 \mathbf{G}_R & \dots & \mathbf{F}_R^{\dim \mathbf{x}-1} \mathbf{G}_R \end{bmatrix}. \quad (32)$$

The rank of the matrix $\mathbf{C}$ is equal to $\dim \mathbf{x}_R$. Because $\operatorname{rank} \mathbf{C} \neq \dim \mathbf{x}$, the dynamic system defined by (29) is not controllable, and the only controllable state variable is the robot pose.

In the context of the partial controllability of the SLAM system shown above, it is interesting to investigate its observability. The observability matrix of (29) is given by

$$\mathbf{O} = \begin{bmatrix} \mathbf{H} & \mathbf{HF} & \mathbf{HF}^2 & \dots & \mathbf{HF}^{\dim \mathbf{x}-1} \end{bmatrix}^T. \quad (33)$$

The rank of the matrix $\mathbf{O}$ is equal to $\dim \mathbf{x} - \dim \mathbf{x}_{F_i}$, where $\mathbf{x}_{F_i}$ is the state vector of a single map feature. Because $\operatorname{rank} \mathbf{O} \neq \dim \mathbf{x}$, the analysed dynamic system is not observable.

There are important practical implications of the partial controllability and observability of the EKF-SLAM system. In a partially controllable dynamic system, the noise $\mathbf{v}$ does not affect all the state components. In the EKF-SLAM system, this is a consequence of the static world assumption—the part of the state vector representing the map has no dynamics. The controllability property prevents the system covariance matrix from being singular (Bar-Shalom *et al.*, 2001). In EKF-SLAM, this property does not hold, so the determinant of that part of the system covariance matrix which is associated to the map states will asymptotically tend to zero. This situation is undesirable, because it leads to optimistic covariance estimates of the map features.

In the EKF-SLAM system, the covariance matrix contains correlations between all state variables. The fully correlated system hinders full observability of the state estimate. Partial observability makes this estimate dependant on the initial state—the robot pose at the time of the first feature observation (Andrade-Cetto and Sanfeliu, 2005). This result coincides with Property 3, which defines a lower bound of the map uncertainty.

4. Extensions to the basic EKF-SLAM algorithm

4.1. Improving the computational efficiency. A disadvantage of EKF-SLAM application is the high computational cost of this algorithm, caused by the complexity of the EKF estimation (correction) stage. The polynomial complexity of this stage in EKF-SLAM is an implication of the assumed representation of the SLAM posterior (1) as a multidimensional Gaussian PDF (7). This PDF is represented by the SLAM state vector and the covariance matrix that contains correlations between all state variables. Thus, to correct the PDF upon new observations, the covariance matrix of size $n \times n$ has to be updated, which means the computation of all the correlations, and requires a number of operations proportional to n^2. Because n (the number of features in the map) grows as the robot explores the environment, the computational complexity makes it difficult to estimate an increasing number of features in real time and may cause problems in an environment large enough. It should be noted that the inter-feature and robot-to-feature correlations (off-diagonal elements of $\mathbf{C}_x$) cannot be simply ignored, because this quickly leads to inconsistencies in the map and optimistic estimates of the features, as was shown in (Castellanos *et al.*, 1997).

Although the actual number of features that can be handled in real-time by EKF-SLAM depends on the environment characteristics and the implementation detail (Skrzypczyński, 2007b), much of the research on EKF-SLAM has been focused on improving the computational efficiency of this algorithm. A common idea in most of these improved EKF-SLAM variants is to divide the stochastic map into some smaller portions.

Some methods delay the incorporation of new observations into the global map while the robot stays inside a confined region of the global map. Only features in a local map around the robot are updated every time a new observation is made, while the global map may be updated at a much lower frequency. The delayed update idea is exploited in the compressed EKF (Guivant and Nebot, 2001), which applies the EKF algorithm to a small set of features in a local area of the map. This decreases the computational cost, since the entire covariance matrix need not be updated. However, when the robot moves outside of this local area, the information from the local map must be propagated to the global one, so the entire system state (8) and the covariance matrix (9) have to be updated. This method yields the same results as the full EKF, but its computational requirements are reduced by a constant factor, because the full map updates are performed occasionally.

There are also modifications of the EKF-SLAM approach that divide the global map into smaller submaps with their local coordinate systems. This allows the EKF-SLAM algorithm to work only on a small amount of fea-

tures at a time, thus the computational requirements can be made independent of the global map size. However, aligning the submaps together to build a global map may be an issue. Some methods, like decoupled stochastic mapping (Leonard and Feder, 2000), use submaps that are anchored to a global coordinate frame. More recent approaches usually use submaps linked through coordinate transformations between their origins. The Atlas approach (Bosse *et al.*, 2003) maintains a graph whose nodes are submaps with local coordinate frames, and edges represent transformations between these frames. A new submap is created whenever the uncertainty of the robot pose grows above some limit. This allows constant time updates with the EKF. However, a separate map matching algorithm must be employed to identify the areas that have been visited before, e.g., when closing loops in the environment.

A similar approach is hierarchical SLAM (Estrada *et al.*, 2005), which uses submaps with local coordinate systems arranged in a hierarchical manner. This method uses the number of features in the local map as the criteria to start a new submap and implements an effective loop closing algorithm. Generally, all relative submap approaches result in a network of optimal submaps estimated with the full EKF-SLAM algorithm and connected by conservative, sub-optimal links. A positive side-effect of using local submaps is the reduction of linearisation errors, because submap-based EKF-SLAM versions apply all EKF updates on confined local maps where the robot orientation error is small.

All the above mentioned solutions to the SLAM problem, although computationally more effective, are based on EKF-SLAM at the core. Thus, they share the most fundamental limitation of the Kalman filtering—assumed Gaussianity of all probability distributions involved in the filter. An alternative to EKF implementation of the recursive Bayes filter concept is the particle filter, which allows estimating non-Gaussian, multimodal PDFs. Although particle filtering proved to be effective in mobile robot self-localization (Fox *et al.*, 2001), a direct application of this algorithm to solve the SLAM problem is computationally infeasible, because the number of particles required to represent a given PDF scales exponentially with the dimensionality of the state space considered, which may be very high in the SLAM problem (every new feature adds a dimension).

An effective method to apply particle filtering in SLAM was proposed by Montemerlo (2003). This algorithm, known as FastSLAM, exploits the idea that whenever the robot path is known, the observation of each feature does not provide information about other features. Thus, given the robot path, each feature is independent of the rest of the features. This allows us to factor the SLAM

posterior (1) into a product of simpler terms:

$$p(\mathbf{x}_R^k, \mathbf{m} \mid \mathbf{z}^k, \mathbf{u}^k)$$

$$= p(\mathbf{x}_R^k \mid \mathbf{z}^k, \mathbf{u}^k) \prod_{i=1}^{n} p(\mathbf{m}_i \mid \mathbf{x}_R^k, \mathbf{z}^k, \mathbf{u}^k), \quad (34)$$

where $\mathbf{x}_R^k = \{\mathbf{x}_{R(1)}, \mathbf{x}_{R(2)}, \ldots, \mathbf{x}_{R(k)}\}$ represents the robot path, $\mathbf{m}_i$ is the i-th point feature (landmark) in the map, and n is the number of features. The posterior distribution $p(\mathbf{x}_R^k \mid \mathbf{z}^k, \mathbf{u}^k)$ is represented by a set of weighted samples (particles) in the particle filter, while the map related to each particle is represented as n independent, two-dimensional Gaussian distributions.

The estimation is decoupled into a particle filter that estimates the robot path and n landmark location estimators, which are implemented as independent EKFs for each particle. This decoupling of the SLAM problem into a self-localization problem and a mapping problem with uncorrelated features makes it possible to decrease the computational complexity of the algorithm to $O(m \log n)$, where m is the number of particles. FastSLAM has proved to be effective even for large outdoor environments (Montemerlo, 2003), but some concerns have been raised about its tendency to diverge over time (Bailey *et al.*, 2006).

4.2. Effective pairing of features and observations.
In spite of many improvements, data association is still a challenge in EKF-SLAM. Incorrect associations between features and observations easily lead to EKF divergence. Whenever feature matching fails, and a re-observed feature is not associated with its counterpart stored in the map, a spurious feature is introduced to the map, which may cause false pairings in further time steps, eventually leading to EKF divergence. Moreover, if the number of spurious features is high, then the global map grows quickly, making the SLAM computational complexity problem worse.

Because the matching procedure is external with respect to the EKF itself, the uncertainty of matching cannot be represented by the filter. The spatial uncertainty of features is assumed to be Gaussian (i.e., unimodal), which makes it impossible to represent alternative hypotheses about feature locations in the map. Therefore, once an observation and a feature are paired, and the system state gets updated, this decision is irreversible, even if further observations suggest that it was wrong. Also, the robot pose is represented as a single Gaussian PDF. Thus, only one hypothesis about the robot pose at a time is maintained by the filter. As a result, EKF-SLAM represents the uncertainty of the robot pose but cannot represent its ambiguity, which can arise due to perceptual aliasing, e.g., in an environment with substantial symmetry.

The most popular observation-to-feature matching strategy in various EKF-SLAM implementations is the

nearest neighbor filter, mostly due to its implementational simplicity and low computational complexity, which is linear in the number of map features. However, the statistical gating on the likelihood of the observation given map and robot pose estimates is effective in choosing the right pairing only if the probability of the right association is higher than the probability of matching an observation with an unrelated map feature. To keep the right association probability high, the feature observation uncertainty, the feature location uncertainty, and the robot pose estimate uncertainty all have to be smaller than the separations between the environment features. Unfortunately, in real-world applications, these properties do not hold whenever the environment is cluttered, less precise sensors are used (e.g., monocular vision), the observations are ambiguous (a typical case for sonars), or the robot pose uncertainty is large, e.g., when revisiting a mapped area after a traversal of a long loop. So, the success of the nearest neighbor filter depends on both the quality of the observations and the local characteristics of the environment, and thus it is hardly predictable.

More advanced data association strategies have been proposed, such as Joint Compatibility Branch and Bound (JCBB) (Neira and Tardós, 2001) and the Combined Constraint Data Association (CCDA) (Bailey, 2002). The authors of JCBB observed that the innovations in the matchings of different observations obtained from the same vehicle position are correlated. Hence, their method considers simultaneously multiple associations and provides the largest set of feature correspondences that are jointly consistent. The branch and bound algorithm is employed to efficiently traverse the tree of possible joint correspondences in search of the hypothesis that maximize the number of jointly compatible pairings. By choosing the hypothesis with the largest consensus, the robustness of matching is significantly improved. The CCDA algorithm is based on a similar idea; however, a graph of data association constraints is constructed instead of a tree of joint correspondences. Each node in this graph represents a pairing between an observation and a map feature. Edges in this graph represent joint compatibility between pairs of data associations. The algorithm finds the set of jointly compatible pairs that correspond to the largest clique in the graph.

Although batch-validation methods like JCBB and the CCDA enable more robust statistical data association, they eventually all choose a single data association hypothesis to be used in the EKF-SLAM correction stage. Thus, the data association step is still irreversible. The only possibility to make it reversible with uni-modal Kalman filter PDFs is to explicitly maintain multiple data association hypotheses over time. This approach is known as Multiple Hypothesis Tracking (MHT) in the target tracking domain (Bar-Shalom *et al.*, 2001). If multiple data associations pass the validation gate (16)

for a particular observation, a new hypothesis is created for each possible association. The resulting hypotheses are then split again whenever new data associations are ambiguous, which results in exponential growth in the number of hypotheses that have to be maintained. To keep the number of hypotheses under control, a probability value can be assigned to each hypothesis, and then less probable hypotheses can be pruned by using some heuristics. Although the MHT approach has been successfully used for mobile robot self-localization with feature-based maps (Austin and Jensfelt, 2000), so far it has not been applied in EKF-SLAM. The reason is the computational cost: while in self-localization each hypothesis is just a robot pose, in EKF-SLAM it should be an entire stochastic map, which makes this approach infeasible.

4.3. Perceptually rich environment representations.
Robust data association becomes essential when the robot has to correct a larger pose error, e.g., during loop closing. In such a case, statistical data association is usually insufficient. Most of the successful EKF-SLAM implementations to date employ laser scanners that provide precise and reliable range measurements. However, to relocalize itself, the robot should recognize unambiguously its surroundings, which in many environments is impossible relying solely on geometry and 2D laser data, due to environment symmetries and the lack of distinguishable geometric features at the height of the laser beam plane.

Recently, vision has received much attention in SLAM research, because of its ability to capture a rich description of the environment, including both geometric and photometric information. The greatest advantage of computer vision seems to be the ability to recognize complex features such as posters on walls, door plates, etc., which are highly salient and unambiguous. However, passive visual sensing has many limitations. Particularly, the popular and simple monocular vision systems provide only bearing information without an indication of range to the detected feature. There are also some issues in visual feature extraction that are specific to the context of feature-based SLAM: the detected visual features are usually initialized with a much bigger positional uncertainty than their geometric counterparts detected by a laser, and the number of visual features included in the stochastic map of a typical indoor environment is much bigger than the number of line segments describing this environment. These issues lead to data association problems (a map cluttered with highly uncertain features), and limit visual EKF-SLAM scalability (Davison *et al.*, 2007).

Laser scanners and vision sensors, particularly monocular vision, have quite orthogonal and complementary properties. The 2D laser scanner provides a small amount of highly accurate geometric data, which are, however, quite ambiguous within the context of place recognition and data association. On contrary, a monoc-

ular vision system enables to extract rich photometric description of the surroundings useful in data association but lacks the ability to precisely locate the detected features. Hence, it is meaningful to use laser data to estimate the location of the features and to use the vision data to make these features more distinguishable in the data association process.

Within the context of EKF-SLAM, a multi-sensor system consisting of a 2D laser scanner and a camera was employed in (Castellanos and Tardós, 1999). In this approach, the idea of multi-sensor fusion was exploited in a classic sense, improving the accuracy of the resulting features but adding to them no additional discriminative power that stems from photometric information.

An entirely different approach to perceptually rich environment representation in SLAM was presented in (Ortin *et al.*, 2004). The idea is to some extent a combination of the feature-based and appearance-based approaches to SLAM. Laser scans are segmented into vertical planes containing grey level textures obtained by the camera and mapped from the images to the planes by means of a homography. Stored rich textures allow the planes to be recognized unambiguously, thus providing a solution to the first-location problem in SLAM.

Newman and Ho (2005) showed how monocular vision data can be used for loop closing in SLAM. In their work, visually salient, affine invariant feature descriptors extracted from images are stored in a database. Information about the stored descriptors matching the current visual perception is used to discover loop closing events. Vision-based loop closing is integrated into a scan-matching-based SLAM algorithm.

In the recent work (Skrzypczyński, 2008), we presented new methods to increase the discriminative properties of the laser-based geometric features used in EKF-SLAM by employing monocular vision data. The 2D laser scanner is our primary sensor yielding a precise geometric description of the environment in the form of 2D horizontal line segments. A reasonable assumption for an office-like environment is that these segments represent vertical planar surfaces (walls, pieces of furniture, etc.). An advantage of line segments and other features having physical dimensions is their better mutual discriminance, due to unary geometric constraints imposed by the dimensions. Unfortunately, segment features are seldom observed in their full length, due to occlusions and the limited range of laser scanners.

The length of a partially observed segment is not a reliable constraint because it depends on the spatial uncertainty of the segment endpoints, which in turn depends on the laser scan segmentation method, and has a weak support in the sensor model (Skrzypczyński, 2007a). The length can be treated as a constraint only if the segment is delimited by some vertical features, such as corners or door frames. Taking this into account, we add to the line
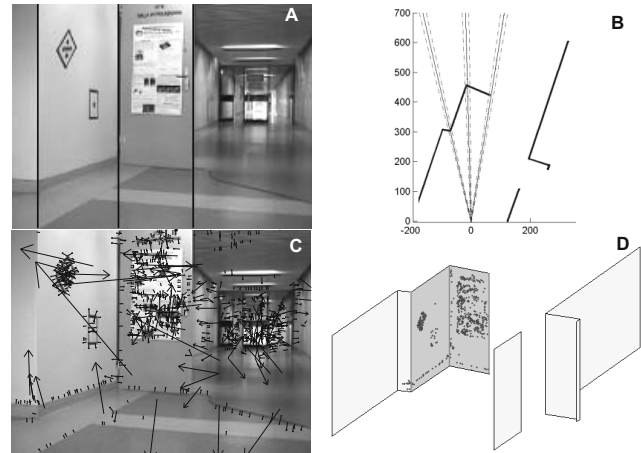

Fig. 3. Example of a local map built with the proposed method.

segments reliable geometric constraints on their length by fusing the horizontal lines with vertical edges extracted from monocular images. These constraints and the assumption that segments are parts of planar vertical surfaces allow us to convert the segments into semiplanes and embed into them descriptors of salient photometric features. Such photometric information can give the segments much more distinctiveness and enable robust place recognition in SLAM.

In order to efficiently embed the photometric information into the segments, we need a method of extracting distinctive features from images. One of the most effective methods to extract various photometric features is the Scale Invariant Feature Transform (SIFT) (Lowe, 2004). The SIFT algorithm detects interest keypoints, which are local extremes of difference-of-Gaussian images through the location and scale space. An SIFT descriptor vector is computed for each processed region. This descriptor is invariant to image translation, rotation, scaling and partially invariant to illumination changes. The SIFT keypoints found in an image are projected onto the vertical semiplanes extracted from the corresponding laser scan by using projective geometry and camera-scanner calibration data. This is done under the assumption that the photometric features represented by SIFT vectors are located on approximately flat vertical surfaces. Because the sensors provide no reliable information about the height of the vertical semiplanes, we set this height to the typical height of walls in the experimental site. A data structure consisting of a 2D segment limited in length by photometry-confirmed vertical edges and a set of SIFT descriptors located inside the rectangular area created by this segment and the edges is considered a new feature type, called the Perceptually Rich Segment (PRS).

Figure 3 shows the results of local map building by fusing the geometric and photometric information extracted from a laser scan and a camera image taken at
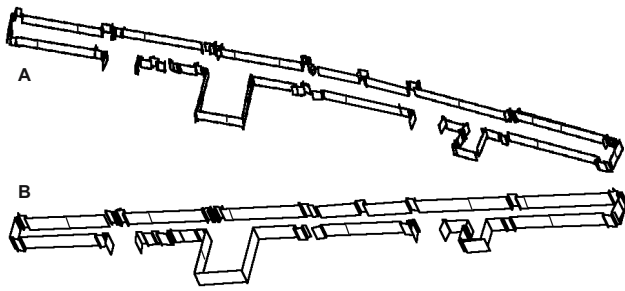
Fig. 4. Examples of global maps built with segment-based (a) and PRS-based (b) representations

the same robot pose. In Fig. 3(a) strong vertical edges found on the image are shown. These edges are depicted in Fig. 3(b) as vision edges with respect to the reference frame of the laser scanner overlaid on the segments extracted from the scan. The dashed lines indicate the spatial uncertainty of these features. SIFT keypoints extracted from the image are visualized in Fig. 3(c). The resulting local map consisting of "plain" segment features (light grey) and PRS features (dark grey) is shown in Fig. 3(d). Note that the SIFT keypoints embedded in the existing PRSs clearly represent visually salient features in this scene, particularly the poster on the doors and the two landmarks attached to the wall.

Figure 4 shows a more extended experiment performed in a corridor, to investigate how the proposed PRS features improve the quality of the segment-based global map. In Fig. 4(a), a map built from the laser data gathered during this experiment is depicted. This map, built by the EKF-SLAM system as described in (Skrzypczyński, 2007a), is of reasonable quality. However, it contains a number of artifacts due to erroneous termination of segments. Long walls have been broken into several overlapping shorter segments due to false matches of geometric features. The whole map is also slightly bent. Simple data association became insufficient because of a large uncertainty in the vehicle pose caused by maneuvers in a corridor lacking enough geometric features perpendicular to the direction of motion.

The results obtained in the same experiment but with an EKF-SLAM system using both the laser and camera data and employing the PRS features (when possible) are shown in Fig. 4(b). The map is more consistent. The vision-verified vertical edges correctly terminated the segments that were problematic in the laser-only case. Compared with the geometric-only case, the photometric features (vertical edges on the walls) contributed to the reduction of the uncertainty in the robot pose, enabling more robust feature matching.

5. Conclusions

The last decade has brought many significant advancements in SLAM research, including new computationally effective SLAM algorithms, major applications of the known methods, and deeper understanding of the problem structure and properties. In spite of these achievements, there is still no single SLAM algorithm that can be termed a "standard solution". All known approaches to SLAM have their own limitations, and require careful implementation, customized to the particular environment and robot characteristics. However, there is a consensus among researchers that probabilistic problem formulation and solution should be taken, because of the crucial role played in SLAM by rigorous treatment of the spatial uncertainty.

This paper provides an introduction to probabilistic, feature-based SLAM, and a survey of the state-of-the-art EKF-SLAM versions and extensions. Moreover, new techniques are proposed to remedy some of the known problems in EKF-SLAM: (i) the role of reliable orientation estimation in keeping EKF-SLAM consistent is pointed out, and it is shown that a scan matching procedure can be used to accomplish this task; (ii) the use of photometric features to increase the discriminative properties of line segments commonly used in indoor EKF-SLAM is demonstrated.

References

Andrade-Cetto, J. and Sanfeliu, A. (2005). The effects of partial observability when building fully correlated maps, *IEEE Transactions on Robotics* **21**(4): 771–777.

Andrade-Cetto, J., Vidal-Calleja, T. and Sanfeliu, A. (2005). Unscented transformation of vehicle states in slam, *Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain*, pp. 323–328.

Arras, K. O., Tomatis, N., Jensen, B. T. and Siegwart, R. (2001). Multisensor on-the-fly localization: Precision and reliability for applications, *Robotics and Autonomous Systems* **34**(2–3): 131–143.

Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing* **50**(2): 174–187.

Austin, D. and Jensfelt, P. (2000). Using multiple Gaussian hypotheses to represent probability distributions for mobile robot localization, *Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA*, pp. 1036–1041.

Bailey, T. (2002). *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*, Ph.D. thesis, University of Sydney, Sydney.

Bailey, T., Nieto, J., Guivant, J., Stevens, M. and Nebot, E. (2006). Consistency of the EKF-SLAM algorithm, *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems, Beijing, China*, pp. 3562–3567.

Bailey, T., Nieto, J. and Nebot, E. (2006). Consistency of the FastSLAM algorithm, *Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, FL, USA*, pp. 424–429.

Bar-Shalom, Y., Li, X. R. and Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation*, Wiley, New York, NY.

Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W. and Teller, S. (2003). An atlas framework for scalable mapping, *Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan*, pp. 1899–1906.

Castellanos, J. A., Neira, J. and Tardós, J. D. (2004). Limits to the consistency of the EKF-based SLAM, *Preprints of the IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal*, (on CD-ROM).

Castellanos, J. A. and Tardós, J. D. (1999). *Mobile Robot Localization and Map Building. A Multisensor Fusion Approach*, Kluwer, Boston, MA.

Castellanos, J. A., Tardós, J. D. and Schmidt, G. (1997). Building a global map of the environment of a mobile robot: The importance of correlations, *Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, USA*, pp. 1053–1059.

Davison, A. J., Reid, I. D., Molton, N. D. and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6): 1052–1067.

DiMarco, M., Garulli, A., Giannitrapani, A. and Vicino, A. (2004). A set-theoretic approach to dynamic robot localization and mapping, *Autonomous Robots* **16**(1): 23–47.

Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, H. F. and Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem, *IEEE Transactions on Robotics and Automation* **17**(3): 229–241.

Estrada, C., Neira, J. and Tardós, J. D. (2005). Hierarchical SLAM: Real-time accurate mapping of large environments, *IEEE Transactions on Robotics* **21**(4): 588–596.

Fox, D., Thrun, S., Burgard, W. and Dellaert, F. (2001). Particle filters for mobile robot localization, *in* A. Doucet (Ed.), *Sequential Monte Carlo Methods in Practice*, Springer, Berlin, pp. 499–516.

Gasós, J. and Rosetti, A. (1999). Uncertainty representation for mobile robots: Perception, modeling and navigation in unknown environments, *Fuzzy Sets and Systems* **107**(1): 1–24.

Guivant, J. and Nebot, E. (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation, *IEEE Transactions on Robotics and Automation* **17**(3): 242–257.

Huang, S. and Dissanayake, G. (2006). Convergence analysis for extended Kalman filter based slam, *Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, FL, USA*, pp. 412–417.

Julier, S. J. and Uhlmann, J. K. (2001). A counter example to the theory of simultaneous localization and map building, *Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, South Korea*, pp. 4238–4243.

Kozłowski, K. and Pazderski, D. (2004). Modelling and control of a 4-wheel skid-steering mobile robot, *International Journal of Applied Mathematics and Computer Science* **14**(4): 477–496.

Leonard, J. J. and Feder, H. J. S. (2000). A computationally efficient method for large-scale concurrent mapping and localization, *Robotics Research: The Ninth International Symposium*, Springer, London, pp. 169–179.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60**(2): 91–110.

Maybeck, P. S. (1979). *Stochastic Models, Estimation, and Control*, Academic Press, New York, NY.

Montemerlo, M. (2003). *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.

Neira, J. and Tardós, J. D. (2001). Data association in stochastic mapping using the joint compatibility test, *IEEE Transactions on Robotics and Automation* **17**(6): 890–897.

Newman, P. and Ho, K. (2005). SLAM-loop closing with visually salient features, *Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain*, pp. 644–651.

Ortin, D., Neira, J. and Montiel, J. M. M. (2004). Relocation using laser and vision, *Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA*, pp. 1505–1510.

Paz, L., Piniés, P., Tardós, J. D. and Neira, J. (2008). Large-scale 6-dof SLAM with stereo in hand, *IEEE Transactions on Robotics* **24**(5): 946–957.

Skrzypczyński, P. (2005). Uncertainty models of the vision sensors in mobile robot positioning, *International Journal of Applied Mathematics and Computer Science* **15**(1): 73–88.

Skrzypczyński, P. (2006). Uncertain spatial knowledge management in a mobile robot architecture, *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, Heidelberg, Germany*, pp. 420–425.

Skrzypczyński, P. (2007a). *Perception Uncertainty Management in a Mobile Robot Navigation System*, Dissertations, No. 407, Poznań University of Technology Press, (in Polish).

Skrzypczyński, P. (2007b). Spatial uncertainty management for simultaneous localization and mapping, *Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy*, pp. 4050–4055.

Skrzypczyński, P. (2008). Fusing laser and vision data for perceptually rich environment description, *Maintenance Problems* (3): 57–67.

Smith, R., Self, M. and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics, *in* G. W. I. Cox (Ed.), *Autonomous Robot Vehicles*, Springer, Berlin, pp. 167–193.

Piotr Skrzypczyński graduated from the Poznań University of Technology (1993). He received the Ph.D. and D.Sc. degrees in robotics from the same university in 1997 and 2007, respectively. Since 1998 he has been an assistant professor at the Institute of Control and Information Engineering (ICIE) of the Poznań University of Technology, and the head of the Mobile Robotics Laboratory of the ICIE. Dr. Skrzypczyński is the author or co-author of over 90 technical papers in the fields of robotics and computer science. His current research interests include autonomous mobile robots, simultaneous localization and mapping, multisensor fusion, distributed robotic systems, and computational intelligence methods in robotics.