Robotics and Autonomous Systems

Hybrid simultaneous localization and map building: a natural integration of topological and metric

Nicola Tomatis[†][*], Illah Nourbakhsh[‡], Roland Siegwart[†]

[†]*Autonomous Systems Lab, Swiss Federal Institute of Technology Lausanne (EPFL), CH 1015 Lausanne, Switzerland*

[‡]*The Robotics Institute, Carnegie Mellon University (CMU), 5000 Forbes Avenue, Pittsburgh, PA 1513, USA*

Abstract

In this paper the metric and topological paradigms are integrated in a hybrid system for both localization and map building. A global topological map connects local metric maps, allowing a compact environment model, which does not require global metric consistency and permits both precision and robustness. Furthermore, the approach handles loops in the environment during automatic mapping by means of the information of the multimodal topological localization. The system uses a 360° laser scanner to extract corners and openings for the topological approach and lines for the metric method. This hybrid approach has been tested in a 50 x 25 m^2 portion of the institute building with the fully autonomous robot Donald Duck. Experiments are of four types: maps created by a complete exploration of the environment are compared to estimate their quality; test missions are randomly generated in order to evaluate the efficiency of the approach for both the localization and relocation; the fourth type of experiments shows the practicability of the approach for closing the loop.

Keywords: Mobile robot navigation; Hybrid (metric - topological); Kalman filtering; POMPD

1. Introduction

Research in localization and automatic mapping has recently lead to successful approaches. However, solutions for consistent mapping allowing precise and robust localization in unmodified, dynamic, real-world environments have not yet been found. The problem is highly complex due to the fact that it requires the robot to remain localized with respect to the portion of the environment which has already been mapped in order to build a coherent map.

Current research has diverged to different approaches: metric, topological or hybrid navigation schemes have been proposed and studied. Approaches using purely metric maps [17], [8] and [15] are vulnerable to inaccuracies in both map-making and odometry abilities of the robot. Even by taking into account all relationships between features and the robot itself, in large environments the drift in the odometry makes the global consistency of the map difficult to maintain [6]. Landmark-based approaches, which rely on the topology of the environment [13] can better handle this problem, because they only have to maintain topological global consistency, not metric. However these approaches are either less precise than fully metric approaches [16], [5] and [11], due to the discretization of the localization space, or computationally intractable for fully autonomous robots, when fine-grained grids are used [10]. More recently, approaches combining the topological and the metric paradigm [18], [19] and [3] have shown that positive characteristics of both can be integrated to compensate for the weakness of each single approach.

This paper proposes a natural integration of both the metric and topological paradigms to combine the best characteristics of both universes. For this, the model embodies both a metric and a topological representation. The metric model consists of infinite lines that belong to the same place. These places are related to each other by means of a topological map which is composed of nodes representing topological locations

[*] Corresponding author.
E-mail address: n.tomatis@ieee.org (N. Tomatis).

and edges between nodes. Connections between a node and a place are a special case: travelling along these edges causes a switch from the topological to the metric paradigm. The effectiveness of this method for localization has already been shown in [20]. In this paper an extension to automatic mapping which permits the handling of loops in the environment is presented.

The metric approach is an *Extended Kalman Filter* (EKF). This method has already proven its strength for localization [2]. Map building can then be done with the *Stochastic Map* approach [17]. Topological navigation uses a *Partially Observable Markov Decision Process* (POMDP) [5] for state estimation. This permits efficient planning in the large, has an advantageous symbolic representation for man-machine interaction and is robust due to its multi-hypothesis tracking.

2. Environment modeling

The environment is described by a global topological map, which permits moving in the whole environment, and local metric maps which can be used by the robot as soon as it needs further localization precision (see also fig. 1). The only requirement specific to this model is to have a *detectable metric feature* when travelling from a topological node to a metric place. This permits the system to determine the transition point where the change from topological to metric has to be executed and allows robust initialization of the metric localization (i.e. relocation). Given this metric feature local metric maps can be placed anywhere in the environment.

Switching to topological does not require any specific characteristic: the robot navigates metrically to the initialization position for the current local place where it resumes its topological navigation.

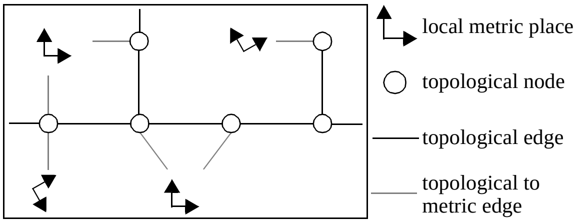


Fig. 1. The environment is represented by places given by their metric maps and nodes representing topological locations. When travelling from a node to a place, the system switches from topological to metric and vice-versa.

2.1. Global topological map

Landmarks, which are helpful for the topological model, are those discriminating between locations in the environment. In this case two different types are chosen:

- Corners, characterized by their orientation.
- Openings, that are also used for model transition.

The topological map can be viewed as a graph. Topological locations are represented by nodes containing the information about the way to reach the connected topological location / metric place. Furthermore, the landmarks lying between two locations are represented as a list between the two nodes. In fig. 2 the graph representing the topological model is viewed for a portion of the environment.

The corner extractor returns a set of (x, y, θ) parameters in robot coordinates, representing the position and orientation of the corners with respect to the robot. Furthermore an extraction confidence parameter p_c is calculated for each corner by taking into account its size. Openings are either large steps perpendicular to the direction of motion in hallways or transitions from rooms to hallways. They can either be a transition between a hallway and a room or between two perpendicular hallways. Due to the use of a 360° laser scanner, an observation contains many landmarks which are transformed in a graph compatible to the environment model, as shown in fig. 3.

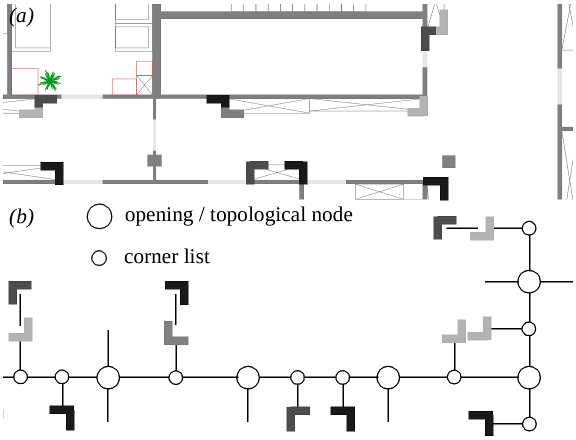


Fig. 2. (a) A portion of an hallway with the extracted corner and opening features. (b) The topological map is represented by a graph. It contains nodes connected to each other with the list of corner features lying between them. Openings (topological nodes) can either be a transition to a room or be a connection to another hallway.

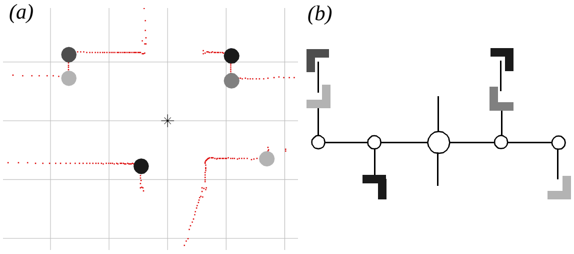


Fig. 3. (a) Laser data and the extracted features. (b) The resulting observation graph.

2.2. Local metric maps

The features used for metric environmental representation are infinite lines. They are less informative than line segments, but have a better probabilistic model with analytical solution and permit a very compact representation of structured geometric environments requiring only about 10 bytes per m^2 for a typical office environment. In fig. 4 a typical office is shown with the lines used for its local metric map. The line model is $\rho\cos(\varphi-\alpha)-r = 0$, where (ρ, φ) is the raw measurement and (α, r) the model parameters. α is the angle of the perpendicular to the line, r its length. The extraction algorithm used has been described in [1]. Its result is a set of (α, r) parameters with their 2 x 2 covariance matrix, which is calculated by propagating the uncertainty from the laser measurements.

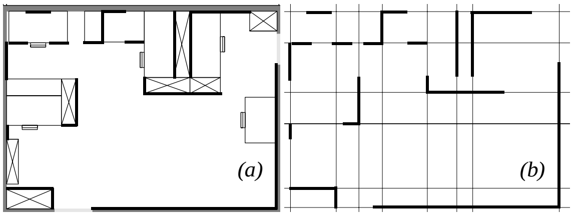


Fig. 4. An office of the institute (a) and the lines representing it in the local metric map (b). The black segments permit to see the correspondence between the two figures.

3. Localization and map building

The environment models allow the use of two different navigation methods with complementary characteristics. The metric localization permits a very precise positioning at the goal point [2], [20] whereas the topological one [5], [20] guarantees robustness against getting lost due to the multimodal representation of the robot's location.

3.1. Map building strategy

As explained in section 2., the environment model is composed of a global topological map and a set of local metric maps. Given a metric transition feature, local metric maps can be everywhere in the environment. Even if the approach is applicable to any structured environment, a suitable environment-dependent strategy has to be adopted.

For many possible application scenarios it can be expected that the robot will have to be very precise in rooms, where most of its tasks have to be executed (e.g. docking for power recharging; manipulation tasks with objects on a table; human-robot interaction). While navigating in the large (i.e. hallways), precision with respect to the features is less important, but robustness and global consistency take an important role. Because of this, the two different levels of abstraction are used in combination of the different type of environmental structures:

- While navigating in hallways the robot firstly creates and then updates the global topological map
- When it enters a room, it creates a new local metric map

These two environmental structures are recognized by means of the laser sensor: thin and long open spaces are assumed to be hallways, while other open spaces will be defined as rooms.

3.2. Exploration strategy

The proposed exploration strategy is simple: the robot first explores all the hallways in a depth-first way. It then explores each room it encountered by backtracking. Note that, in general, for each hallway the room exploration reduces to a linear list traversal. Rooms with multiple openings cause two special cases, which are treated in the next paragraphs.

Rooms with an opening to another room: The robot continues building the current metric map. This leads to the next case if the other room has an opening to a hallway.

Rooms with multiple openings to a hallway: Due to the metric navigation mode during room exploration, the robot knows the direction of the opening and can therefore deduce if it opens to the same hallway, a known one or a new one. In the case of known hallways, the robot simply goes back to the hallway it was coming from and continues its exploration. This could result in two metric maps for the same metric place, one for each opening. In the case of a new hallway, the exploration continues in a hallway depth-first way.

3.3. Topological localization and map building

The current experimental test bed is a part of the institute building. This environment is rectilinear and mainly composed of offices, meeting rooms and hallways. Therefore, only four directions of travel are employed: N, E, S, W. However, this is not an inherent loss of generality because it is not a general requirement of the POMDP algorithm.

Position estimator: Given a finite set of environment states *S*, a finite set of actions *A* and a state transition model *T*, the model can be defined by introducing partial observability. This includes a finite set *O* of possible observations and an observation function *OS*, mapping *S* into a discrete probability distribution over *O*. $T(s, a, s')$ represents the probability that the environment makes a transition from state s to state s' when action a is taken. $OS(o, s, a)$ is the probability of making an observation o in state s after having taken action a. The probability of being in state s' (*belief state* of s') after having made observation o while performing action a is then given by the equation:

$$SE_{s'}(k+1) = \frac{OS(o, s', a)\sum_{s \in S} T(s, a, s')SE_s(k)}{P(o|a, SE(k))} \qquad (1)$$

where $SE_s(k)$ is the belief state of s for the last step, $SE(k)$ is the belief state vector of last step and $P(o|a, SE(k))$ is a normalizing factor. The observation function *OS* is made robust by the fact that an observation is composed of many landmarks (fig. 3), rising its distinctiveness. When no openings are visible, $T(s, a, s) = 0.99$ while $T(s, a, s') = 0.01$ for $s \neq s'$. When the robot encounters an opening, the most probable state s' is searched by comparing the travelled distance *d*, measured starting from *s*, to the information saved in state node *s* during map building. In this case $T(s, a, s') = 0.99$ while $T(s, a, s") = 0.01$ for $s" \neq s'$.

Heading estimator: Because the position estimator does not take into account the heading of the robot, this is done separately as in [11]. However in this case the orientation is estimated by a weighted mean of each observed line that is either horizontal or vertical with respect to the environment. The success of this method is guaranteed by the fact that, in general, lines given by the environmental structures are either parallel or perpendicular to the direction of travel. Infinite lines are matched by means of the validation test

$$(z^{[i]} - \hat{z}^{[j]})\, S_{ij}^{-1}\, (z^{[i]} - \hat{z}^{[j]})^T \le \chi^2_{\alpha,n} \qquad (2)$$

where prediction $\hat{z}^{[j]}$ is directly the odometry state vector variable θ and $\chi^2_{\alpha,n}$ is a number taken from a χ^2 distribution with $n = 1$ degrees of freedom. This can be viewed as an EKF for heading only, where no map is required because for prediction θ is directly used instead.

Control strategy: Since it is computationally intractable to compute the optimal POMDP control strategy for a large environment [5], simple suboptimal heuristics are introduced. For the system presented here the *most likely state* policy has been adopted: the world state with the highest probability is found and the action that would be optimal for that state is executed. However it can happen that the robot is not sure about its current state. This is calculated by mean of the unconfidence function $U(SE(k))$, which is the entropy of the probability distribution over the states of the map. The POMDP is confident when

$$U(SE(k)) = -\sum_s SE_s(k)\log SE_s(k) < U_{max} \qquad (3)$$

where U_{max} is determined by experience. When the robot is unconfident, it follows the hallway in the direction where it expects to find more information.

Map building: Instead of using a complex scheme for model learning like in [12] and [19], where an extension of the *Baum-Welch algorithm* is adopted, here the characteristics of the observation graph (fig. 3) are used. When the robot feels confident about its position, it can decide if an extracted landmark is new by comparing the observation graph to the node in the map corresponding to the most likely state. This can happen either in an unexplored portion of the environment or in a know portion, where new landmarks appear due to the environment dynamic. As explained in section 2.1., the landmarks have an extraction confidence p_l. This characteristic is firstly used to decide if the new landmark can be integrated in the map. When an opening landmark is extracted, it is integrated in the map as a new state node (fig. 2) with a rough measure of the distance to the last state node.

Furthermore, for each integrated landmark, the confidence p_l is used to model the probability of seeing that landmark the next time p_{lmap}. When it is re-observed, the probability in the map is averaged with the confidence of the extracted one. If the robot does not see an expected landmark the probability $1 - p_{lmap}$ is used instead.

$$p_{lmap}(t_i) = \sum_{i=1}^{n} \frac{p_l(t_i)}{n} \qquad (4)$$

$$\text{where} \, p_l(t_i) = \begin{cases} p_l(t_i), & observed \\ 1 - p_{lmap}(t_{i-1}), & \neg observed \end{cases} \qquad (5)$$

When the confidence p_{lmap} decreases and is below a minimum, the corresponding landmark is deleted from the map. This allows for dynamics in the environment, where landmarks that disappear in the real world will be deleted from the map too.

3.4. Model transition

Because the topological navigation method is multimodal, the confidence before switching to the unimodal metric navigation is very critical. In contrast to pure topological navigation, a false state estimate when switching to metric would cause the robot to be in a false place when using the map describing the goal position. If such a problem occurs a solution for detecting this situation and exiting the current local place would be required in order to allow the robot to relocate itself by means of the topological approach. To limit all this, in the current implementation, a door find and passing action is executed only when the estimator is confident (see eq. (3)).

When switching from topological to metric the Kalman filter has to be initialized (i.e. $\hat{x}(k|k)$ and $P(k|k)$ are unknown). This *relocation* problem can be simplified for this approach. As explained in chapter 2., detectable metric feature (door in this case) between a node and a place permits knowing when to switch and gives an approximation of the robot position with respect to the local metric map. The first two moments of the measure are used to initialize the Kalman filter and permit a fast convergence of the filter.

Changing from metric to topological reduces to a metric navigation to the initialization position of the current local place and the resume of the POMDP.

3.5. Closing the loop

The problem of *closing the loop* can be defined as the question of how to know when a location has already been explored, meaning that the environment contains a loop and that the loop in the map must also be closed. In [19] this is achieved by adding a topological mapper which ensures global consistency. This information is then used to correct the global metric map which eventually converges to a global consistent map.

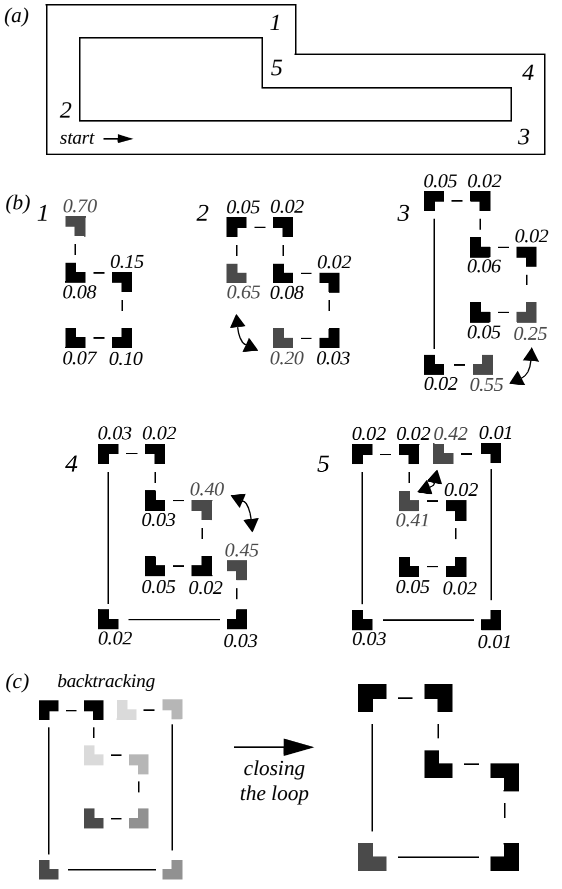


Fig. 5. (a) A loop in the environment. (b) Mapping with the POMDP. 1) The map when the robot is at position 1 in the environment. 2) The robot is re-exploring the start point. The observation function $OS(o, s, a)$ gives high values for both the new node in the map and for the start node, but the probability distribution has not yet diverged because the transition function $T(s, a, s')$ gives a low probability of coming at the map start. 3) However, by moving in the same way on the map the distribution diverges and the POMDP becomes unconfident. 4) The distribution has diverged and the two peaks move in the same way to 5. (c) The mapping is stopped. The loop is closed by backtracking.

The current approach differs in two main aspects:

- Instead of closing the loops only by means of the perception, loops are detected and closed by means of the *localization information*.
- Loops have to be closed only in the topological map because the metric model is represented by many disconnected local metric maps.

Loops can also exist in a local metric map; however, in such small maps the drift in odometry between two updates should not cause any problem to the local consistency, as it has been shown in [6].

The current method works as follows: the robot does not try to recognize if a single observation has already been seen somewhere else. However, as soon as the robot creates the map for a part of the environment which has already been visited, the probability distribution starts diverging into two peaks: one for the current map position; another for the previously created location representing the same physical place. The algorithm starts tracking the two highest probabilities as soon as the POMPD becomes unconfident because this is the first clue indicating a divergence of the probability distribution. A loop can then easily be detected when the distribution has converged into two peaks which move in the same way. The position where the loop has to be closed can be detected by turning off the automatic mapper and backtracking with localization until the distribution re-converges to a single peak. This should be the point where the robot started mapping the loop. An example is given in fig. 5.

3.6. Metric localization and map building

This section briefly describes the main characteristics of the *Stochastic Map* approach [17], which permits using an *Extended Kalman Filter* [8], [15] for localization.

With this approach both the robot position $x_r = (x, y, \theta)'$ and the features $x_i = (\alpha, r)'$ are represented in the system state vector:

$$x = \begin{bmatrix} x_r \\ x_1 \\ \\ x_n \end{bmatrix} \quad C(x) = \begin{bmatrix} C_{rr} & C_{r1} & \dots & C_{rn} \\ C_{1r} & C_{11} & \dots & C_{1n} \\ \dots & \dots & \dots & \dots \\ C_{nr} & C_{n1} & \dots & C_{nn} \end{bmatrix} \tag{6}$$

This represents the uncertain spatial relationship between objects in the map, which is changed by three actions:

- Robot displacement
- Observation of a new object
- Re-observation of an object already existing in the map

Robot displacement: When the robot moves with an uncertain displacement $\boldsymbol{u}$ given by its two first moments (u, C_u), which are measured by the odometry, the robot state is updated to $g(x_r, \boldsymbol{u})$. The updated position and uncertainty of the robot pose are obtained by error propagation on *g*:

$$x_r(k+1) = g(x_r(k), u) = x_r(k) \oplus u \tag{7}$$

$$C_{rr}(k+1) = G \begin{bmatrix} C_{rr}(k) & C_{ru}(k) \\ C_{ur}(k) & C_u \end{bmatrix} G^T \tag{8}$$

where ⊕ is compounding operator and *G* is the Jacobean of *g* with respect to x_r and $\boldsymbol{u}$.

New object: When a new object is found, a new entry must be made in the system state vector. A new row and column are also added to the system covariance matrix to describe the uncertainty in the object's location and the inter-dependencies with the other objects. The new object $(\hat{x}_{new}, C_{new})$ can be integrated in the map by computing the following equations of uncertainty propagation:

$$x_{N+1}(k) = g(x_r(k), x_{new}) = x_r(k) \oplus x_{new} \tag{9}$$

$$C_{N+1N+1}(k) = G_{x_r} C_{rr}(k) G_{x_r}^T + G_{x_{new}} C_{new} G_{x_{new}}^T \tag{10}$$

$$C_{N+1i}(k) = G_{x_r} C_{ri}(k) \tag{11}$$

Re-observation: Let x_{new} be the new observation in the robot frame. The measurement equation is defined as:

$$z = h(x_r, x_{new}, x_i) = g(x_r, x_{new}) - x_i \tag{12}$$

x_{new} is temporarily included in the state to apply the EKF. However, if prediction x_i satisfies the validation test

$$(x_{new} - x_i)\, S_{newi}^{-1}\, (x_{new} - x_i)^T \le \chi^2_{\alpha,n} \tag{13}$$

where $S_{newi} = C_{newnew} + C_{ii} - C_{newi} - C_{inew}$, $\chi^2_{\alpha,n}$ is a number taken from a χ^2 distribution with $n = 2$ degrees of freedom and α the level on which the hypothesis of pairing correctness is rejected, then x_{new} is a re-observation of x_i.

Extended Kalman Filter: When a spatial relationship is re-observed, the updated estimate is a weighted average of the two estimates calculated by means of an

EKF. It permits to update a subset of the state vector while maintaining the consistency by means of the covariance matrices. A measurement equation $z = h(x_1, x\ldots x_m)$ is considered as a function of m relationships included in x. All of the n estimates x_i of the state vector x are updated by a value which is proportional to the difference $\delta = z - \hat{z}$ between the ideal measurement z and the actual measurement $\hat{z}$:

$$x_i(k+1) = x_i(k) + \Gamma_{iz}\Gamma_{zz}^{-1}\delta \tag{14}$$

$$\Gamma_{iz} = E[x_i\delta^T] = \sum_{j=1}^{M} C_{ij}H_{xj}^T \tag{15}$$

$$\Gamma_{zz} = E[\delta\delta^T] = \sum_{j=1}^{M}\sum_{k=1}^{M} H_{x_j}C_{jk}H_{x_k}^T \tag{16}$$

where H_{x_j} is the Jacobean matrix of h with respect to x_j.

The variance and covariance C_{ij} are also updated:

$$C_{ij}(k+1) = C_{ij}(k) - \Gamma_{iz}\Gamma_{zz}^{-1}\Gamma_{jz}^T \tag{17}$$

4. Experimental results

The approach has been tested in the 50 x 25 m^2 portion of the institute building shown in fig. 7 with four different types of experiments for a total of more than 1.5 km.

For the experiments, Donald Duck has been used (fig. 6). It is a fully autonomous mobile vehicle running XO/2, a deadline driven hard real-time operating system [4]. Donald navigates locally by means of a motion control algorithm, which plays the role of both position control and obstacle avoidance: it reaches the given (x, y, θ) or (x, y) goal by planning a collision free path (with respect to the current local data), and reacting to the dynamic environment either by merely replanning the path or by changing heading direction and replanning when an object appears in front of the robot.

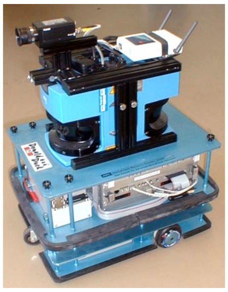


Fig. 6. The fully autonomous robot Donald Duck. Its controller consists of a VME standard backplane with a Motorola PowerPC 604 microprocessor clocked at 300 Mhz running XO/2. Among its peripheral devices, the most important are the wheel encoders, a 360° laser range finder and a grey-level CCD camera.

4.1. Map building

In this section the automatic mapping capabilities of the presented approach are evaluated. Note that the environment is arbitrarily closed (fig. 7), so that the exploration procedure is finite. Furthermore local metric maps are taken from the *a priori* map used in [2], because the *stochastic map* is not yet implemented on the robot and runs, therefore, only off-line.

For this evaluation, five maps generated by complete explorations of the environment shown in fig. 7 are compared to evaluate their quality with respect to consistency and completeness. In order to evaluate the topological mapper first, maps are compared before the backtracking step. By knowing which door is open during the exploration, it can be extrapolated how many state nodes should be extracted (see the black dots in fig. 7). Their position (odometry) and type (opening or hallway) are stored during exploration to check whether the resulting model is consistent with the real environment. For the other features (corners), each resulting map is compared to the others to calculate the average amount of differences between a pairs of maps. The results are presented in table 1.

Number of explorations	5
Total travelled distance	343 m
Number of states in the environment	13
Mean detected states	12.8 / 98%
Mean confused hallway/opening	1.2 / 9.2%
Mean detected features	78
Mean different features	18 / 23%

Table 1: Comparison of five maps generated by complete explorations of the environment shown in fig. 7.

One of the problems encountered during the exploration is the difficulty of distinguishing between openings and hallways. This leads to a mean of 1.2 false detections for each experiment. Nevertheless by visiting all the openings when traversing the environment by backtracking to add the local metric maps, these errors are detected and corrected. In one experiment a state (opening) was not extracted at all.

For the corner features it is more difficult to define which features really exist in the environment. What is easy to see is the difference between two maps. The mean amount of extracted corners in a map is 78; an average of 18 of these are noisy features that are not always extracted. This means that 77% of the features are constant in the five maps showing that the perception delivers valuable information to the mapper.

4.2. Localization

The quality of a map can also easily be estimated by testing it for localization. For this, two types of localization experiments are performed: one for localization (position tracking) and the other for relocation.

To test the topological localization, 25 randomly generated test missions for a total of about 900 m and 28000 estimates are performed. The robot knows in which state it is at the start point. A mission is successful when the robot reaches its goal location, is in front of the opening and is confident about its position. There it switches to the metric approach. To have more information about the experiments, each state transition is stored in a log file with all the information permitting to determine if each state transition detected by the localization took place physically. The results are presented in table 2. Even if all the missions are successful the log file permits to detect 21 false state transitions that caused 404 false estimates in *B* and *B'* (fig. 7), where the peak probability moved forward and backward between two neighbor states. These false estimates represent only 1.4% of the total, meaning that the system recovers quite fast from these errors. Nevertheless the robot had also confident false estimates (0.5%) that can cause a mission failure if the goal state is estimated when the robot is in front of a another opening.

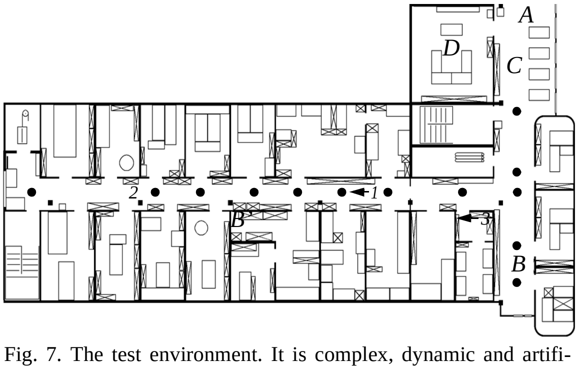


Fig. 7. The test environment. It is complex, dynamic and artificially closed in A so that the exploration procedure is finite. Black dots are the places where the automatic mapper is expected to extract state nodes (the other doors are closed). In B and B' the robot had problems distinguishing between the two neighbour locations. C and D are detected as rooms and represented by a single local metric map. A large loop does not exist in this environment. Therefore, for the experiments in section 4.3., a loop is "artificially created" by starting the exploration in 1, stopping it in 2, taking the robot manually to 3 and resuming.

Number of missions	25
Success	25 / 100%
Total travelled distance	899 m
Mean travel distance	36 m
Mean travel speed	0.31 m/s
Total real state transitions	181
False state transitions	21 / 12%
Total estimates	27870
Unconfident states	3413 / 12%
False estimates	404 / 1.4%
Confident false estimates	149 / 0.5%

Table 2: Localization experiments. All the test missions have been successfully performed. However the robot also made false state transitions that caused some false estimates (1.4%). This happened only by B and B' in fig. 7. The reason that lead to a success rate of 100% is that the system always recovered from its error without estimating the goal location in front of a false

The second type of test is focused on recovering from a lost situation (relocation). Ten experiments are started from a randomly defined position in the environment with a uniform belief state distribution (i.e. lost situation). The goal is to measure which distance or amount of state transitions are required in order to converge to a correct confident state estimate. To avoid false interpretations, the robot is required to travel 3 state nodes further without estimate errors to fulfill the test. In table 3 the ten tests are briefly resumed.

As expected the robot can always recover. Its policy is simple: go forward until recovery or end of hallway; If end of hallway, turn. The system requires a minimum of 1 and a maximum of 4 states to recover. The interesting point is that this difference in the results is position dependent and repeatable. For example the crossing between the two hallways permits recovery with a single state transition because it is global distinctive for the environment in fig. 7. On the other hand, the right part of the horizontal hallway seems to be more distinctive than the left one where the robot require the maximum amount of states to recover.

Number of experiments	10
Total travelled distance	250 m
Mean distance for recovering	13.7 m
Min / max distance for recovering	1.21 / 20.31 m
Mean number of state for recovering	2.11
Min / max state for recovering	1 / 4

Table 3: Recovering from a lost situation (i.e. overall constant belief state). The robot requires from 1 to 4 states to recover, depending on the distinctiveness of the part of the environment where it is moving.

The metric localization is used but not explicitly tested here, because the used EKF has already been extensively tested in [2] with a total of 6.4 km. The mean 2σ-error bounds are approximately 1 centimeter in x and y and 1 degree for θ. Furthermore the metric localization approach has also been tested with this hybrid method for localization on the same robot in [20], where ground truth measurements at goal position resulted in an average error of less than 1 cm.

4.3. Closing the loop

In the test environment there are no large loops. In order to test the proposed approach a loop is artificially created by displacing the robot during the exploration as shown in fig. 7. As explained in section 3.5., it can be assumed that when two peaks appear and move in the same way for three subsequent state transitions a loop has been discovered. In all the other experiments this has effectively never appeared, showing that this a good test for loops. This experiment has been performed three times. Each time the probability distribution has effectively diverged into two peaks allowing the detection of the loop. In order to close the loop the robot has turned off the mapping algorithm and has gone back until the distribution has converged to a single confident peak. This took place where the map has been started *(1* in fig. 7) proving that the loop could be closed correctly.

5. Related work

Successful navigation of embedded systems for real applications relies on the precision that the vehicle can achieve, the capacity of not getting lost and the practicability of their algorithms on the limited resources of the autonomous system. Furthermore the fact that *a priori* maps are rarely available and, even when given, not in the format required by the robot, and that they are mainly unsatisfactory due to imprecision, incorrectness and incompleteness, makes automatic mapping a real need for application-like scenarios.

Simultaneous localization and map building research can be divided into two main categories: metric and topological. Metric approaches are defined here as methods, which permit the robot to estimate its (x, y, θ) position, while topological are those where the position is given by a location without precise metric information.

After the first precise mathematical definition of the *stochastic map* [17], early experiments [8], [15], have shown the quality of fully metric simultaneous localization and map building: the resulting environment model permits highly precise localization, which is only bounded by the quality of the sensor data [2]. However these approaches suffer of some limitations. Firstly they rely strongly on odometry. For automatic mapping this makes the global consistency of the map difficult to maintain in large environments, where the drift in the odometry becomes too important. Furthermore they represent the robot pose with a single Gaussian distribution. This means that an unmodeled event (i.e. collision) could cause a divergence between the ground-truth and the estimated pose from which the system is unable to recover (lost situation). In [6] it has been shown that by taking into account all the correlations (off-diagonal cross-covariances in eq. 6), the global consistency is better maintained. However this is not sufficient, as confirmed by a recent work [7], where a solution is proposed by extending the absolute localization to include a localization relative to local frames.

On the other hand topological approaches [13] can handle multi-hypothesis tracking and have a topological global consistency, which is easier to maintain. The robustness of such approaches has firstly been proven by the application of the *state set progression* [16], which has then been generalized to the POMDP approach [5], [11]. For automatic mapping in [12] the *Baum-Welch algorithm* has been used for model learning. In contrast to the above mentioned topological approaches, [14] proposes a topological approach, which heavily rely on odometry in order better to handle environment dynamics. All these approaches are robust, but have the drawback of losing in precision with respect to the fully metric ones: the robot pose is represented by a location without precise metric information. To face this, *Markov localization* [10] has been proposed: a fine-grained grid guarantees both precision and multimodality. However, this approach remains computationally intractable for current embedded systems. A more efficient alternative has recently been proposed, but the *Monte Carlo localization* [9] has not yet been extended for simultaneous localization and mapping.

Metric and topological approaches are converging, like [7], [9] and [10], to hybrid solutions by adding advantageous characteristics of the opposite world. Going in this direction, in [18] the approach consists of extracting a topological map from a grid map by means of a *Voronoi* based method, while [19] proposes to use the *Baum-Welch algorithm* as in [12], but to build a topologically consistent global map which permits closing the loop for the global metric map too. In [3] a discrete *Markov Model* is used to generate hypotheses, which are then tracked by multiple *Kalman* trackers.

In contrast to the above mentioned approaches, for this system a natural integration of the metric and topological paradigm is proposed. The approaches are completely separated into two levels of abstraction. Metric maps are used only locally for structures (rooms) that are naturally defined by the environment. There, a fully metric method is adopted. As it has been shown in [6], for such small environments, where the drift in the odometry remains uncritical, *stochastic map* allows for precise and consistent automatic mapping. The topological approach is used to connect the local metric maps that can be far away from each other. With this the robot can take advantage of the precision of a fully metric EKF navigation added to the robustness in the large of the POMDP approach. All this by maintaining a compactness of the environment representation and a low complexity, which allows an efficient implementation of the method on a fully autonomous system. This hybrid approach shows also its practicability for environments with loops. In this case the loop is closed in the global topological map based on the information from the topological localization, while the metric information remains local and does therefore not require further processing, contrasting to [19], where the topological information is used for mapping only, to close the loop in the metric map correctly.

6. Conclusions and outlook

This paper presents a hybrid approach for both localization and map building. The metric and topological parts are completely separated into two levels of abstraction. Together they allow a very compact and computationally efficient representation of the environment for mobile robot navigation. Furthermore this combination permits both precision with the non-discrete metric estimator and robustness by means of the multimodal topological method.

The approach is validated empirically by extensive experimentation for a total of more than 1.5 km. Map building is tested by performing five complete explorations of the environment and comparing the resulting maps. This comparison demonstrates that the maps are consistent with respect to the environment and that the perception permits to extract precious information. For localization, the success rate over the 0.9 km of the 25 tests missions is 100%. Nevertheless a precise analysis of the state transitions shows that, between neighbor states, false state estimate occurs (1.4%) and sometimes are even treated as confident (0.5%). The relocation performance of the topological method has been shown with 10 successful experiments where the belief state converges with 1 to 4 state transitions depending on the distinctiveness of the part of the environment where the robot is navigating. It has been shown how loops can be closed on the localization level instead of the perception level. This is easily done by using the multi-hypothesis tracking characteristic of the POMDP for detection and backtracking for closing the loop.

These experiments show that the presented approach is mature and appropriate for real applications in office-like environments. Further research will therefore focus on long term experimentation in large indoor environments presenting other different and challenging characteristics. Nevertheless, the problem of switching from topological to metric remains a non neglectable limitation to the generality of the approach. A solution facing this limitation has to be implemented as proposed in section 3.4.

References

[1] K. O. Arras, R.Y. Siegwart, Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building, in: Proceedings of SPIE, Mobile Robotics XII, Vol. 3210, 1997.

[2] K. O. Arras, N. Tomatis, B. Jensen, R. Siegwart, Multisensor On-the-Fly Localization: Precision and Reliability for Applications, Robotics and Autonomous Systems **34** (2001).

[3] H. Baltzakis, P. Trahanias, Hybrid Mobile Robot Localization using Switching State-Space Models, in: Proceedings of the IEEE International Conference on Robotics and Automation, Washington DC, USA, 2002.

[4] R. Brega, N. Tomatis, K. Arras, The Need for Autonomy and Real-Time in Mobile Robotics: A Case Study of XO/2 and Pygmalion, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan, 2000.

[5] A. R. Cassandra, L. P. Kaelbling, et al., Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation, in: Proceedings of the IEEE International Conference on Robotics and Automation, Osaka, Japan, 1996.

[6] J. A. Castellanos, J. D. Tardós, et al., Building a Global Map of the Environment of a Mobile Robot: The Importance of Correlations, in: Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, 1997.

[7] J. A. Castellanos, M. Devy, et al., Simultaneous Localization and Map Building for Mobile Robots: A Landmark-based Approach, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, 2000.

[8] J.L. Crowley, World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging, in: Proceedings of the IEEE International Conference on Robotics and Automation, Scottsdale, AZ, 1989.

[9] F. Dellaert, D. Fox et al., Monte Carlo Localization for Mobile Robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, 1999.

[10] D. Fox, Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation, Institute of Computer Science III, Bonn, Germany, University of Bonn, 1998.

[11] R. Gutierrez-Osuna, R. C. Luo, LOLA: probabilistic navigation for topological maps." AI Magazine **17**(1), 1996.

[12] S. Koenig, R.Goodwin, et al., Robot Navigation with Markov Models: A Framework for Path Planning and Learning with Limited Computational Resources, in: Proceedings of the International Workshop, Reasoning with Uncertainty in Robotics, Amsterdam, Netherlands, Springer, 1995.

[13] B. J. Kuipers, Y. T. Byun, A qualitative approach to robot exploration and map-learning, in: Proceedings of the Workshop on Spatial Reasoning and Multi-Sensor Fusion, Los Altos, CA, USA, Morgan Kaufmann, 1987.

[14] C. Kunz, T. Willeke et al., Automatic Mapping of Dynamic Office Environments, Autonomous Robots **7**, 1999.

[15] J.J. Leonard, H.F. Durrant-Whyte, et al., Dynamic Map Building for an Autonomous Mobile Robot, The International Journal of Robotics Research **11**(4), 1992.

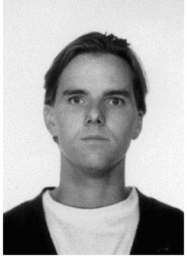
[16] I. Nourbakhsh, Dervish: An Office-Navigating Robot. Artificial Intelligence and Mobile Robots. D. Kortenkamp, R. P. Bonasso and R. Murphy, The AAAI Press/ The MIT Press**:** 73-90, 1998.

[17] R. C. Smith, M. Self, et al., Estimating uncertain spatial relationships in robotics, Uncertainty in Artificial Intelligence 2, J. F. Lemmer and L. N. Kanal, Elsevier Science Pub.: 435-461, 1988.

[18] S. Thrun, A. Bücken, Integrating grid-based and topological maps for mobile robot navigationIn Proceedings of the National Conference on Artificial Intelligence, Portland, OR, USA, 1996.

[19] S. Thrun, J.-S. Gutmann, et al., Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach, in: Proceedings of the Tenth Conference on Innovative Applications of Artificial Intelligence, Madison, WI, 1998.

[20] N. Tomatis, I. Nourbakhsh, K. Arras, R. Siegwart, A Hybrid Approach for Robust and Precise Mobile Robot Navigation with Compact Environment Modeling, in: Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, Korea, 2001.



Nicola Tomatis (1973) received his M.Sc. in computer science in 1998 at the Swiss Federal Institute of Technology (ETH) Zurich. In the same year he worked as Research Assistant in mobile robotics and robot vision at the Institute of Robotics, ETH. After this, he moved to the Swiss Federal Institute of Technology (EPF) Lausanne, where he received his Ph.D. at the end of 2001. His research covered metric and topological (hybrid) mobile robot navigation, computer vision and sensor data fusion. Since end of 2001 he has a part time position as senior researcher with the Autonomous Systems Lab, EPF, where he leads the BIBA European Commission project for the lab, he is responsible for the software management and continues his research in hybrid navigation, sensor data fusion, man-machine interaction and robot safety and robustness. On the other hand he is executive board member and leads the software R & D of the BlueBotics SA spin-off, which is involved in mobile robotics.

Illah R. Nourbakhsh is an Assistant Professor of Robotics in The Robotics Institute at Carnegie Mellon University. He received his Ph.D. in computer science from Stanford University in 1996. He is co-founder of the Toy Robots Initiative at The Robotics Institute. His current research projects include electric wheelchair sensing devices, robot learning, theoretical robot architecture, believable robot personality, visual navigation and robot locomotion. His past research has included protein structure prediction under the GENOME project, software reuse, interleaving planning and execution and planning and scheduling algorithms. At the Jet Propulsion Laboratory he was a member of the New Millennium Rapid Prototyping Team for the design of autonomous spacecraft. He is a founder and chief scientist of BluePumpkin Software, Inc. and Mobot, Inc.

Roland Siegwart (1959) received his M.Sc. ME in 1983 and his Doctoral degree in 1989 at the Swiss Federal Institute of Technology (ETH) Zurich. After his Ph.D. studies he spent one year as a postdoc at Stanford University where he was involved in micro-robots and tactile gripping. From 1991 to 1996 he worked part time as R&D director at MECOS Traxler AG and as a lecturer and deputy head at the Institute of Robotics, ETH. During this time he was mainly involved in magnetic bearings, mechatronics and micro-robotics. Since 1997 he is a full professor for Autonomous Systems and Robots at the Swiss Federal Institute of Technology, Lausanne (EPFL), leading a research group of around 20 peoples. His current research interests are robotics and mechatronics, namely high precision navigation, space robotics, human-robot interaction, all terrain locomotion and micro-robotics. He lectures various courses in robotics, mechatronics and smart product design at the two Swiss Federal Institutes of Technology and is cofounder of several spin-off companies. Roland Siegwart published more than 80 papers and is member of various scientific committees. He namely represents Switzerland in the International Federation of Robotics (IFR) and the Advisory Group for Automation and Robotics (AGAR) of the European Space Agency (ESA). He was program and general chair for various international conferences, including the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems.