

Combining Topological and Metric: A Natural Integration for Simultaneous Localization and Map Building

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

[†]Autonomous Systems Lab
Swiss Federal Institute of Technology Lausanne (EPFL)
CH-1015 Lausanne
nicola.tomatis@epfl.ch

[‡]The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 1513
illah@ri.cmu.edu

Abstract

In this paper the metric and topological paradigm are integrated in a single 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 permits to handle loops in the environment by automatic mapping using the information of the multimodal topological localization. The system uses a 360 degree 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 m² portion of the institute building with the fully autonomous robot Donald Duck. Experiments are of three 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 localization approach; The third type of experiments shows the practicability of the approach for closing the loop.

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 been found yet. 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 that 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 are vulnerable to inaccuracies in both map-making and dead-reckoning 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. Landmark-based approaches, which rely on the topology of the environment, 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, due to the discretization of the localization

space, or computationally intractable for fully autonomous robots, when fine grained grids are used. More recently, approaches combining the topological and the metric paradigm (mainly grid-based) 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 maximize the effectiveness of the presented approach by combining 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 that is composed of nodes representing topological locations and edges between nodes. Connections between a node and a place are a special case: Traveling 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 [19]. In this paper it is extended to include an automatic mapping approach, which permits to cope with loops in the environment.

For the metric approach an *extended Kalman filter* (EKF) is used. This method has already proven its strength for localization in [2]. Map building can therefore be done with the *stochastic map* approach [16]. Topological navigation uses a *partially observable Markov decision process* (POMDP) [4] 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 the robot can use as soon as it needs further localization precision (see also fig. 1). In order to switch from topological to metric, a detectable metric feature is needed to determine the transition point and to initialize the metric localization (i.e. relocation). This is the only specific requirement for the presented hybrid approach. Given this transition feature, a metric place can be defined everywhere in the environment.

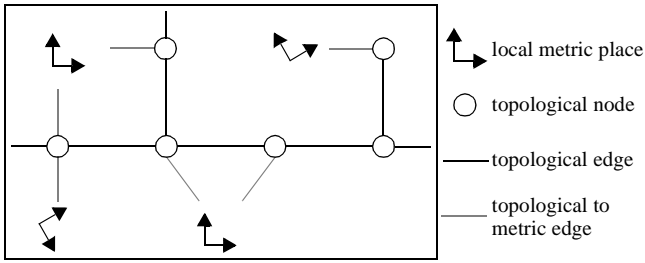


Figure 1: The environment is represented by places given by their metric maps and nodes representing topological locations. When travelling from a node to a place, the system switches from topological to metric and vice-versa.

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

2.1 Global Topological Map

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

- Corners, which are characterized by their orientation.
- Openings, which 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 list of the landmarks lying between two locations is 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

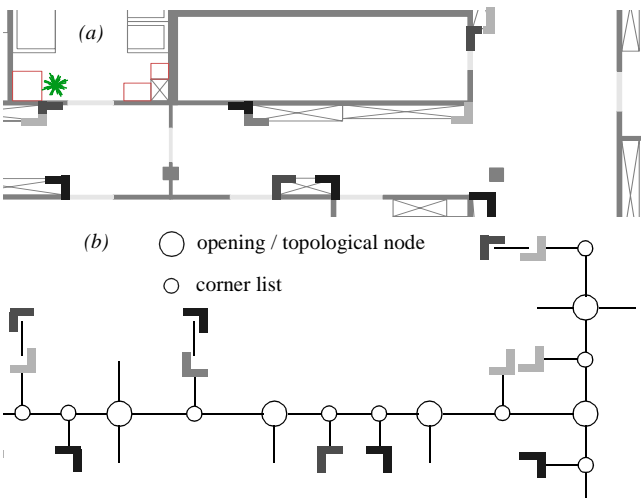


Figure 2: (a) A portion of an hallway with the extracted corner and opening features. (b) The topological map is represented by a graph. It contains nodes connected to each other with the list of corner features lying between them. Openings (topological nodes) can either be a transition to a room or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

extraction confidence parameter p_c is calculated for each corner. The confidence is directly given by the size of the extracted feature. 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 an hallway and a room or between two perpendicular hallways. Note that, because the sensor used is a 360 degree laser scanner, an observation contains many landmarks which are transformed in a graph compatible to the environment model, as shown in fig. 3.

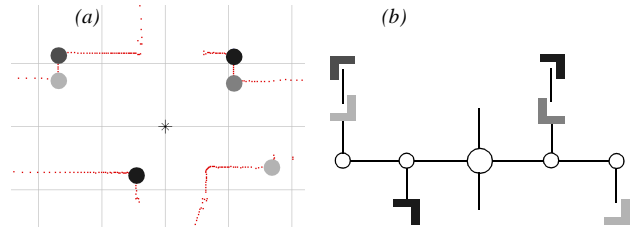


Figure 3: (a) Laser data and the extracted features. (b) The resulting observation graph.

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 used extraction algorithm has been described in [1]. Its result is a set of (α, r) parameters with their 2×2 covariance matrix, which is calculated by propagating the uncertainty from the laser measurements.

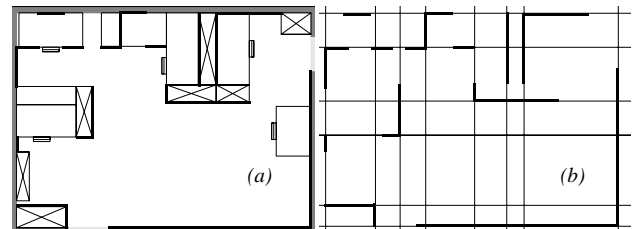


Figure 4: An office of the institute (a) and the lines representing it in the local metric map (b). The black segments permit to see the correspondence between the two figures.

3. Localization and Map Building

The environment models allow the use of two different navigation method with complementary characteristics. The metric localization permits a very precise positioning at the goal point [2], [19] whereas the topological one [4], [19] guarantees robustness against getting lost due to the multi-modal 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 differentiated 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 opening to another room: In this case 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 opening 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 cause having 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 limitation is not an inherent loss of generality because it is not a general requirement of the 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:

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

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 like in [10]. 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

$$(z^{[i]} - \hat{z}^{[j]}) S_{ij}^{-1} (z^{[i]} - \hat{z}^{[j]})^T \leq \chi_{\alpha, n}^2 \quad (2)$$

where prediction $\hat{z}^{[j]}$ is directly the odometry state vector variable θ and $\chi_{\alpha, n}^2$ 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 [4], 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 unconfident function $U(SE(k))$, which is the entropy of the probability distribution over the states of the map. The POMDP is confident when

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

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. What has to be avoided at any cost is to switch from the multimodal topological navigation to the unimodal metric navigation when the robot is unconfident about its location, otherwise it could enter a false local metric place and therefore be lost. If such a problem occurs a solution for detecting this situation and exit the current local place would be required in order to allow the robot relocating itself by means of the topological approach.

Map Building: Instead of using a complex scheme for model learning like in [11] and [18], 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 known portion, where new landmarks appear due to the environment dynamic. As explained in section 2.1, the landmarks come with their 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.

$$p_{lmap}(t_i) = \sum_n \frac{p_l(t_i)}{n} \quad (4)$$

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

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 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 has also to be closed (fig. 5).

In [18] 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.

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*.

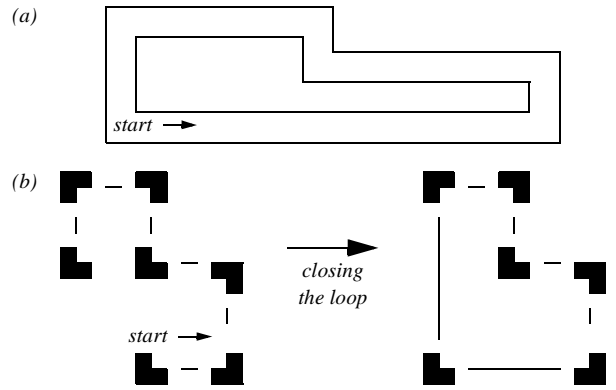


Figure 5: (a) A loop in the environment (b) Its map. Closing the loop in automatic mapping requires the system to know that a location has already been visited.

- Loops have to be closed only in the topological map because the metric model is represented by many disconnect local metric maps.

Loops can also exist in a local metric map, however, due to the fact that these maps are supposed to be small, the drift in odometry does not cause any relevant problem to the local consistency.

The current method works as follows: The robot does not try to recognize if a single perception of the environment 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 to two peaks: One for the position in the map, which is currently being created; Another for the previously created location representing the same physical place. This can easily be detected when the distribution has converged into two peaks. The position where the loop has to be closed can then be detected by backtracking until the distribution re-converges to a single peak. This should also be the start point of the exploration (or a local start point).

3.5 Metric Localization and Map Building

This section describes briefly the main characteristics of the *stochastic map* approach [16], which permits using an *extended Kalman filter* [7], [14] 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:

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

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 \mathbf{u} given by its two first moments (u, C_u), which are measured by the odometry, the robot state is updated to $g(x_r, \mathbf{u})$. The updated position and uncertainty of the robot pose are obtained by error propagation on g :

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

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

where \oplus is compounding operator and G is the Jacobean of g with respect to x_r and \mathbf{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 (x_{new}, C_{new}) can be integrated in the map by computing the following equations of uncertainty propagation:

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

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

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

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

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

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

$$(x_{new} - x_i) S_{newi}^{-1} (x_{new} - x_i)^T \leq \chi_{\alpha, n}^2 \quad (13)$$

where $S_{newi} = C_{newnew} + C_{ii} - C_{newi} - C_{inew}$, $\chi_{\alpha, n}^2$ 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 $z = h(x_1, x, x_m)$ 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 that is proportional to the difference $\delta = z - z_i$ between the ideal measurement z and the actual measurement z_i :

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

$$\Gamma_{iz} = E[x_i \delta^T] = \sum_{j=1}^M C_{ij} H_{x_j}^T \quad (15)$$

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

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

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

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

4. Experimental Results

The approach has been tested in the 50 x 25 m² portion of the institute building shown in fig. 6 with four different types of experiments for a total of more than 1.5 km.

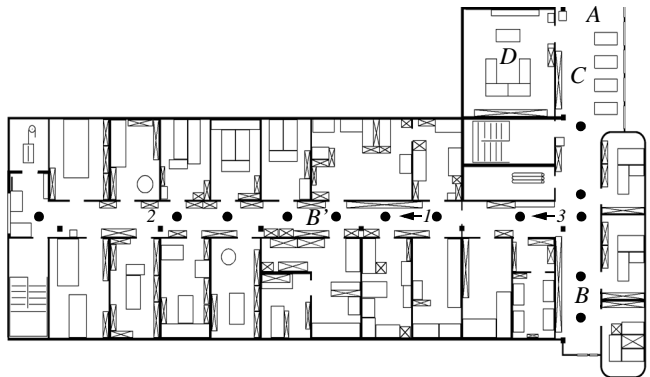


Figure 6: The test environment. It is complex, dynamic and artificially closed in A so that the exploration procedure is finite. Black dots are the places where the automatic mapper is expected to extract state nodes (the other doors are closed). In B and B' the robot had problems distinguishing between 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.

For the experiments, Donald Duck has been used (fig. 7). It is a fully autonomous mobile vehicle running XO/2, a deadline driven hard real-time operating system [3]. Donald navigates locally by means of a motion control algorithm, which plays the role of both position controller 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.

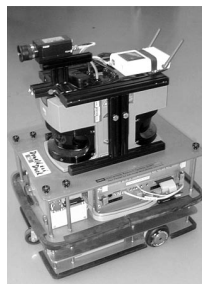


Figure 7: The fully autonomous robot Donald Duck. Its controller consists of a VME standard backplane with a Motorola PowerPC 604 microprocessor clocked at 300 Mhz running XO/2. Among its peripheral devices, the most important are the wheel encoders, a 360° laser range finder and a grey-level CCD camera.

4.1 Map Building

In this section the automatic mapping capabilities of this approach are tested and evaluated. Note that the environment is arbitrarily closed (fig. 6), 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. 6 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. 6). 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 couple 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. 6.

One of the problems encountered during the exploration is the difficulty of distinguishing between opening and hallway. This leads to a mean of 1.2 false detection 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 feature really exists in the environment. What can better be seen 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 almost 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, per-

mitting to know 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. 6), 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 another opening.

| | |
|------------------------------|------------|
| 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. 6. 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 opening. Nevertheless the robot had also confident false estimates (0.5%) that could cause mission failure.

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 an overall constant belief state (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 resumed briefly.

| | |
|-------------------------------------|----------------|
| 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.

As expected the robot can always recover. Its policy is simple: Go forward until recover 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 to recover with a single state because it is global distinctive for the environment in fig. 6. 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.

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 approached the centimeter in x and y and the degree for θ . Furthermore the metric localization approach has also been tested with this hybrid method for localization on the same robot in [19], 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. 6. Because in all the other experiments (map building, localization), except the relocation, no distant twin peaks have appeared, 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. This experiment has been performed three times. Each time the probability distribution has effectively diverged into two peaks allowing to detect the loop. In order to close the loop the robot has gone back until the distribution has converged to a single confident peak. This took place where the map has been started (I in fig. 6) 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 metric information.

After the first precise mathematical definition of the *stochastic map* [16], early experiments [7], [14], 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 dead-reckoning. 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 [5] it has been shown that by taking into account all the correlations (off-diagonal cross-covariance in eq. (6)), the global consistency is better maintained. However this is not sufficient, as confirmed by a recent work [6], where a solution is proposed by extending the absolute localization to include a localization relative to local reference frames.

On the other hand topological approaches [12] 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* [15], which has then been generalized to the POMDP approach [4], [10]. For automatic mapping in [11] the *Baum-Welch algorithm* has been used for model learning. In contrast to the above mentioned topological approaches, [13] proposes a topological approach, which heavily rely on odometry in order better to handle dynamic environments. 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, the *Markov localization* [9] 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* [8] has not yet been extended for simultaneous localization and map building.

Metric and topological approaches are converging, like [6], [8] and [9], to hybrid solutions by adding advantageous characteristics of the opposite world. Going in this direction, in [17] the approach consists in extracting a topological map from a grid map by means of a Voronoi based method, while [18] proposes to use the *Baum-Welch algorithm* as in [11], but to build a topologically consistent global map which permits closing the loop for the global metric map too.

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 [5], 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 [18], 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 new 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 approach.

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 states 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 approach for detection and backtracking for closing the loop.

These experiments show that the presented approach is mature to be introduced for real applications. Further research will therefore focus on long term experimentation in large indoor environments presenting other different and challenging characteristics.

Acknowledgements

Thanks to Kai Arras, Björn Jensen and Gilles Caprari at ASL, EPFL, for their valuable feedback about this work.

References

- [1] Arras, K.O. and R.Y. Siegwart (1997). Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building. Proceedings of SPIE, Mobile Robotics XII, Vol. 3210, p. 42-53.
- [2] Arras, K. O., N. Tomatis, et al. (2001). "Multisensor On-the-Fly Localization: Precision and Reliability for Applications". *Robotics and Autonomous Systems* **34**(2-3).
- [3] Brega, R., N. Tomatis, et al. (2000). The Need for Autonomy and Real-Time in Mobile Robotics: A Case Study of XO/2 and Pygmalion. IEEE/RSJ International Conference on Intelligent Robots and Systems, Takamatsu, Japan.
- [4] Cassandra, A. R., L. P. Kaelbling, et al. (1996). Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation. IEEE International Conference on Robotics and Automation, Osaka, Japan.
- [5] Castellanos, J. A., J. D. Tardós, et al. (1997). Building a Global Map of the Environment of a Mobile Robot: The Importance of Correlations. IEEE International Conference on Robotics & Automation, Albuquerque.
- [6] Castellanos, J. A., M. Devy, et al. (2000). Simultaneous Localization and Map Building for Mobile Robots: A Landmark-based Approach. IEEE International Conference on Robotics and Automation (ICRA), San Francisco.
- [7] Crowley, J.L. (1989). World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. IEEE International Conference on Robotics and Automation, Scottsdale, AZ.
- [8] Dellaert F., D. Fox et al. (1999). Monte Carlo Localization for Mobile Robots. IEEE International Conference on Robotics and Automation, Detroit Michigan.
- [9] Fox, D. (1998). Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation. *Institute of Computer Science III*. Bonn, Germany, University of Bonn.
- [10] Gutierrez-Osuna, R. and R. C. Luo (1996). "LOLA: probabilistic navigation for topological maps." *AI Magazine* **17**(1).
- [11] Koenig S., R. Goodwin, et al. (1995). Robot Navigation with Markov Models: A Framework for Path Planning and Learning with Limited Computational Resources. International Workshop, Reasoning with Uncertainty in Robotics, RUR '95, Amsterdam, Netherlands, Springer.
- [12] Kuipers, B. J. and Y. T. Byun (1987). A qualitative approach to robot exploration and map-learning. Workshop on Spatial Reasoning and Multi-Sensor Fusion, Los Altos, CA, USA, Morgan Kaufmann.
- [13] Kunz, C., T. Willeke et al. (1999). "Automatic Mapping of Dynamic Office Environments." *Autonomous Robots* **7**.
- [14] Leonard, J.J., H.F. Durrant-Whyte, et al. (1992). "Dynamic Map Building for an Autonomous Mobile Robot", *The International Journal of Robotics Research* **11**(4).
- [15] Nourbakhsh, I. (1998). 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.
- [16] Smith R. C., M. Self, et al. (1988). Estimating uncertain spatial relationships in robotics. *Uncertainty in Artificial Intelligence 2*. J. F. Lemmer and L. N. Kanal, Elsevier Science Pub.: 435-461.
- [17] Thrun, S. and A. Bücken (1996). Integrating grid-based and topological maps for mobile robot navigation. National Conference on Artificial Intelligence, AAAI, Portland, OR, USA.
- [18] Thrun, S., J.-S. Gutmann, et al. (1998). Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach. Tenth Conference on Innovative Applications of Artificial Intelligence, Madison, WI.
- [19] Tomatis, N., I. Nourbakhsh, et al. (2001). A Hybrid Approach for Robust and Precise Mobile Robot Navigation with Compact Environment Modeling. IEEE International Conference on Robotics and Automation, Seoul, Korea.