Mobile Robot Mapping and Localization in Non-Static Environments

Cyrill Stachniss   **Wolfram Burgard**

University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany
{stachnis | burgard @informatik.uni-freiburg.de}

Abstract

Whenever mobile robots act in the real world, they need to be able to deal with non-static objects. In the context of mapping, a common technique to deal with dynamic objects is to filter out the spurious measurements corresponding to such objects. In this paper, we present a novel approach to estimate typical configurations of dynamic areas in the environment of a mobile robot. Our approach clusters local grid maps to identify the possible configurations. We furthermore describe how these clusters can be utilized within a Rao-Blackwellized particle filter to localize a mobile robot in a non-static environment. In practical experiments carried out with a mobile robot in a typical office environment, we demonstrate the advantages of our approach compared to alternative techniques for mapping and localization in dynamic environments.

Introduction

Building maps is an essential problem in robotics and has been studied over several years. Most of the approaches to mapping with mobile robots are based on the assumption that the environment is static. As reported by Wang & Thorpe (2002) as well as by Hähnel *et al.* (2002), dynamic objects can lead to serious errors in the resulting map. A popular technique to deal with non-static environments is to identify dynamic objects and to filter out the range measurements reflected by these objects. Whereas such techniques have been demonstrated to be more robust than traditional approaches, their major disadvantage lies in the fact that the resulting maps only contain the static aspects of the environment.

In this paper, we explore an alternative solution to deal with dynamic environments by explicitly modeling the low-dynamic or quasi-static states. Our approach is motivated by the fact, that many dynamic objects appear only in a limited number of possible configurations. As an example, consider the doors in an office environment, which are typically either open or closed. In such a situation, techniques to filter out dynamic objects produce maps which do not contain a single door. This can be problematic since in many
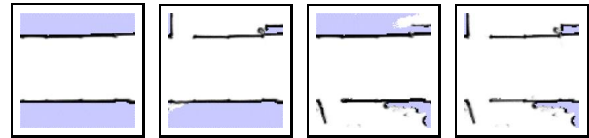
Figure 1: Possible states of a local area. The different configurations correspond to open and closed doors.

corridor environments doors are important features for localization. The knowledge about the different possible configurations can explicitly improve the localization capabilities of a mobile robot. Therefore, it is important to integrate such information into the map of the environment.

As a motivating example consider the individual local maps depicted in Figure 1. These maps correspond to typical configurations of the same place and have been learned by a mobile robot operating in an office environment. They show a part of a corridor including two doors and their typical states. The key idea of our work is to learn such local configurations and to use this information to improve the localization accuracy of a mobile robot.

The contribution of this paper is a novel approach to mapping in low-dynamic environments. Our algorithm divides the entire map into several sub-maps and learns for each of these sub-maps typical configurations of the corresponding part of the environment. This is achieved by clustering local grid maps. Furthermore, we present an extended Monte-Carlo localization algorithm, which uses these clusters in order to simultaneously estimate the current state of the environment and the pose of the robot. Experiments demonstrate that our map representation leads to an improved localization accuracy compared to maps lacking the capability to model different configurations of the environment.

Related Work

In the past, several authors have studied the problem of learning maps in dynamic environments. A popular technique is to track dynamic objects and filter out the measurements reflected by those objects (Hähnel *et al.* 2002; Wang & Thorpe 2002). Enhanced sensor models combined with the EM algorithm have been successfully applied to filter out arbitrary dynamic objects by Hähnel *et al.* (2003). The authors report that filtering out dynamic objects can improve the scan registration and lead to more accurate maps.

Anguelov *et al.* (2002) present an approach which aims to learn models of non-stationary objects from proximity data. The object shapes are estimated by applying a hierarchical EM algorithm based on occupancy grids recorded at different points in time. The main difference to our approach is that we estimate typical configurations of the environment and do not focus on learning geometric models for different types of non-stationary obstacles.

The problem of dealing with people has also been investigated in the context of mobile robot localization. For example, Fox *et al.* (1999) use a probabilistic technique to identify range measurements which do not correspond to a given model. In contrast to our work, they use a fixed, static environmental model and do not reason about configurations the environment can be in. Montemerlo & Thrun (2002) use a method to track walking people while localizing the robot to increase the robustness of the pose estimate.

Romero *et al.* (2001) describe an approach to global localization that clusters extracted features based on similarity. In this way, the robot is able to reduce the number of possible pose hypotheses and can speed up a Markov localization process. The authors also perform a clustering of sub-maps, but compared to our work, they do not consider changes in the environment.

In contrast to most of the approaches discussed so far, we do not address the problem of filtering out or tracking dynamic objects. Our technique is complementary to and can easily be combined with those approaches. We are interested in possible states of the environment, like, e.g., open and closed doors or moved tables. In this context, it makes sense to filter out measurements reflected by walking people, but to integrate those which correspond to obstacles like doors or moved boxes. Our approach learns possible states based on a clustering of local maps. The different environmental state hypotheses enable a mobile robot to more reliably localize itself and to also estimate the current configuration of its surroundings.

In a very recent work, Biber & Duckett (2005) proposed an elegant approach that incorporates changes of the environment into the map representation. Compared to our work, they model temporal changes of local maps whereas we aim to identify the different configurations of the environment.

Learning Maps of Low-Dynamic Environments

The key idea of our approach is to use the information about changes in the environment during data acquisition to estimate possible spatial configurations and store them in the map model. To achieve this, we construct a sub-map for each area in which dynamic aspects have been observed. We then learn clusters of sub-maps that represent possible environmental states in the corresponding areas.

Map Segmentation

In general, the problem of learning maps in dynamic environments is a high-dimensional state estimation problem. A naïve approach could be to store an individual map of the whole environment for each potential state. Obviously, using this approach, one would have to store a number of maps that is exponential in the number of dynamic objects. In real world situations, the states of the objects in one room are typically independent of the states of the objects in another room. Therefore, it is reasonable to marginalize the local configurations of the individual objects.

Our algorithm segments the environment into local areas, called sub-maps. In this paper, we use rectangular areas which inclose locally detected dynamic aspects to segment the environment into sub-maps. For each sub-map, the dynamic aspects are then modeled independently.

Note that in general the size of these local maps can vary from the size of the overall environment to the size of each grid cell. In the first case, we would have to deal with the exponential complexity mentioned above. In the second case, one heavily relies on the assumption that neighboring cells are independent, which is not justified in the context of dynamic objects. In our current system, we first identify positions in which the robot perceives contradictory observations which are typically caused by dynamic elements. Based on a region growing technique, areas which inclose dynamic aspects are determined. By taking into account visibility constraints between regions, they are merged until they do not exceed a maximum sub-map size (currently set to $20\,m^2$). This limits the number of dynamic objects per local map and in this way leads to a tractable complexity. An example for three sub-maps constructed in such a way is depicted in Figure 2. Note that each sub-map has an individual size and different sub-maps can (slightly) overlap.

Learning Environmental Configurations

To enable a robot to learn different states of the environment, we assume that the robot observes the same areas at different points in time. We cluster the local maps built from the different observations in order to extract possible configurations of the environment. To achieve this, we first segment the sensor data perceived by the robot into observation sequences. Whenever the robot leaves a sub-map, the current sequence ends and accordingly a new observation sequence starts as soon as the robot enters a new sub-map. Additionally, we start a new sequence whenever the robot moves through the same area for more than a certain amount of time ($30\,s$). This results in a set Φ of observation sequences for each sub-map

$$\Phi \;\; = \;\; \{\phi_1, \ldots, \phi_n\}, \tag{1}$$

where each

$$\phi_i \;\; = \;\; z_{start(i)}, \ldots, z_{end(i)}. \tag{2}$$

Here z_t describes an observation obtained at time t. For each sequence ϕ_i of observations, we build an individual occupancy grid for the local area of the sub-map. Such a grid is then transformed into a vector of probability values ranging from 0 to 1 and one additional value ξ to represent an unknown (unobserved) cell. All vectors which correspond to the same local area are clustered using the fuzzy k-means algorithm (Duda *et al.* 2001). During clustering, we treat unknown cells in an slightly different way, since we do not

want to get an extra cluster in case the sensor did not covered all parts of the local area. In our experiment, we obtained the best behavior using the following distance function for two vectors a and b during clustering

$$d(a,b) = \sum_i \begin{cases} (a_i - b_i) & a_i \neq \xi \wedge b_i \neq \xi \\ 0 & a_i = \xi \wedge b_i = \xi \\ \epsilon & \text{otherwise,} \end{cases} \quad (3)$$

where ϵ is a constant close to zero.

When comparing two values representing unknown cells, one in general should use the average distance computed over all known cells to estimate this quantity. In our experiments, we experienced that this leads to additional clusters in case a big part of a sub-map contains unknown cells even if the known areas of the maps were nearly identical. Therefore, we use the distance function given in Eq. (3) which sets this distance value to zero.

Unfortunately, the number of different environmental states is not known in advance. Therefore, we iterate over the number of clusters and compute in each step a model using the fuzzy k-means algorithm. In each iteration, we create a new cluster initialized using the input vector which has the lowest likelihood under the current model. We evaluate each model θ using the the Bayesian Information Criterion (BIC) (Schwarz 1978).

$$BIC = \log P(d \mid \theta) - \frac{|\theta|}{2} \log n \quad (4)$$

The BIC is a popular approach to score a model during clustering. It trades off the number $|\theta|$ of clusters in the model θ multiplied by the logarithm of the number of input vectors n and the quality of the model with respect to the given data d. The model with the highest BIC is chosen as the set of possible configurations, in the following also called patches, for that sub-map. This process is repeated for all sub-maps.

Note that our approach is an extension of the classical occupancy grid map (Moravec & Elfes 1985), in which the environment is not supposed to be static anymore. In situations without moving objects, the overall map reduces to a standard occupancy grid map.

The complexity of our mapping approach depends linearly on the number T of observations multiplied by the number s of sub-maps. Furthermore, the region growing applied to build up local maps introduces in the worst case a complexity of $p^2 \log p$, where p is the number of grid cells considered as dynamic. This leads to an overall complexity of $O(T \cdot s + p^2 \log p)$. Using a standard PC, our implementation requires around 20% of the time needed to record the log file.

Monte-Carlo Localization Using Patch-Maps

It remains to describe how our patch-map representation can be used to estimate the pose of a mobile robot moving through its environment. Throughout this paper, we apply an extension of Monte-Carlo localization (MCL), which has originally been developed for mobile robot localization in static environment (Dellaert *et al.* 1998). MCL uses a set of weighted particles to represent possible poses of the robot. Typically, the state vector consists of the robot's position as

well as its orientation. The sensor readings are used to compute the weight of each particle by estimating the likelihood of the observation given the pose of the particle and the map.

Besides the pose of the robot, we want to estimate the configuration of the environment in our approach. Since we do not use a static map like in standard MCL, we need to estimate the map m_t as well as the pose x_t of the robot at time t

$$\begin{aligned} p(x_t, m_t \mid z_{1:t}, u_{0:t-1}) &= \\ &\eta \cdot p(z_t \mid x_t, m_t, z_{1:t-1}, u_{0:t-1}) \\ &p(x_t, m_t \mid z_{1:t-1}, u_{0:t-1}). \end{aligned} \quad (5)$$

Here η is a normalization constant and u_{t-1} refers to the motion command which guides the robot from x_{t-1} to x_t. The main difference to approaches on simultaneous localization and mapping (SLAM) is that we do not reason about all possible map configurations like SLAM approaches do. Our patch-map restricts the possible states according to the clustering of patches and therefore only a small number of configurations are possible.

Under the Markov assumption, the second line of Eq. (5) can be transformed to

$$\begin{aligned} &p(x_t, m_t \mid z_{1:t-1}, u_{0:t-1}) \\ &= \int_{x_{t-1}} \int_{m_{t-1}} p(x_t, m_t \mid x_{t-1}, m_{t-1}, z_{1:t-1}, u_{t-1}) \\ &\quad \cdot p(x_{t-1}, m_{t-1} \mid z_{1:t-1}, u_{0:t-2}) \, dx_{t-1} \, dm_{t-1} \quad (6) \\ &= \int_{x_{t-1}} \int_{m_{t-1}} p(x_t \mid x_{t-1}, m_{t-1}, z_{1:t-1}, u_{t-1}) \\ &\quad \cdot p(m_t \mid x_t, x_{t-1}, m_{t-1}, z_{1:t-1}, u_{t-1}) \\ &\quad \cdot p(x_{t-1}, m_{t-1} \mid z_{1:t-1}, u_{0:t-2}) \, dx_{t-1} \, dm_{t-1} \quad (7) \\ &= \int_{x_{t-1}} \int_{m_{t-1}} p(x_t \mid x_{t-1}, u_{t-1}) p(m_t \mid x_t, m_{t-1}) \\ &\quad \cdot p(x_{t-1}, m_{t-1} \mid z_{1:t-1}, u_{0:t-2}) \, dx_{t-1} \, dm_{t-1}. \quad (8) \end{aligned}$$

Eq. (8) is obtained from Eq. (7) by assuming that m_t is independent from $x_{t-1}, z_{1:t-1}, u_{t-1}$ given we know x_t and m_{t-1} as well as assuming that x_t is independent from $m_{t-1}, z_{1:t-1}$ given we know x_{t-1} and u_{t-1}. Combining Eq. (5) and Eq. (8) leads to

$$\begin{aligned} &p(x_t, m_t \mid z_{1:t}, u_{0:t-1}) \\ &= \eta \cdot p(z_t \mid x_t, m_t, z_{1:t-1}, u_{0:t-1}) \\ &\quad \int_{x_{t-1}} \int_{m_{t-1}} p(x_t \mid x_{t-1}, u_{t-1}) p(m_t \mid x_t, m_{t-1}) \\ &\quad \cdot p(x_{t-1}, m_{t-1} \mid z_{1:t-1}, u_{0:t-2}) \, dx_{t-1} \, dm_{t-1}. \quad (9) \end{aligned}$$

Eq. (9) describes how to extend the standard MCL approach so that it can deal with different environmental configurations. Besides the motion model $p(x_t \mid x_{t-1}, u_{t-1})$ of the robot, we need to specify a map transition model $p(m_t \mid x_t, m_{t-1})$, which describes the change in the environment over time.

In our current implementation, we do not reason about the state of the whole map, since each sub-map would introduce a new dimension in the state vector of each particle, which

leads to a state estimation problem, that is exponential in the number of local sub-maps. Furthermore, the observations obtained with a mobile robot provide information only about the local environment of the robot. Therefore, we only estimate the state of the current patch the robot is in, which leads to one additional dimension in the state vector of the particles compared to standard MCL.

In principle, the map transition model $p(m_t \mid x_t, m_{t-1})$ can be learned while the robot moves through the environment. In our current system, we use a fixed density for all patches. We assume, that with probability α the current state of the environment does not change between time $t-1$ and t. Accordingly, the state changes to another configuration with probability $1 - \alpha$. Whenever a particle stays in the same sub-map between $t-1$ and t, we draw a new local map configuration for that sample with probability $1 - \alpha$. If a particle moves to a new sub-map, we draw the new map state from a uniform distribution over the possible patches in that sub-map. To improve the map transition model during localization, one in principle can update the values for α for each patch according to the observations of the robot. However, adapting these densities can also be problematic in case of a diverged filter or a multi-modal distribution about the pose of the robot. Therefore, we currently do not adapt the values of α while the robot acts in the environment.

Note that our representation bears resemblance with approaches using Rao-Blackwellized particle filters to solve the simultaneous localization and mapping problem (Murphy 1999; Montemerlo *et al.* 2002), as it separates the estimate about the pose of the robot from the estimate about the map. It computes the localization of the vehicle and uses this knowledge to identify the current state of the (local) map. The difference is that we aim to estimate the current state of the sub-map based on the possible configurations represented in our enhanced environmental model.

Experiments

To evaluate our approach, we implemented and thoroughly tested it on an ActivMedia Pioneer II robot equipped with a SICK laser range finder. The experiments are designed to show the effectiveness of our method to identify possible configurations of the environment and to utilize this knowledge to more robustly localize a mobile vehicle.

Application in an Office Environment

The first experiment has been carried out in a typical office environment. The data was recorded by steering the robot through the environment while the states of the doors changed. To obtain a more accurate pose estimation than the raw odometry information, we apply a standard scan-matching technique. Figure 2 depicts the resulting patch-map. For the three sub-maps that contain the doors whose states were changed during the experiment our algorithm was able to learn all configurations that occurred. The sub-maps and their corresponding patches are shown in the same figure.

Figure 2: A patch-map representing the different configurations learned for the individual sub-maps in a typical office environment.

Localization and Sub-Map State Estimation

The second experiment is designed to illustrate the advantages of our map representation for mobile robot localization in non-static environments compared to standard MCL. The data used for this experiment was obtained in the same office environment as above. We placed a box at three different locations in the corridor. The resulting map including all patches obtained via clustering is depicted in Figure 3. Note that the tiles in the map illustrate the average over all patches. To evaluate the localization accuracy obtained with our map representation, we compare the pose estimates to that of a standard MCL using an occupancy grid map as well as a grid map obtained by filtering out dynamic objects (Hähnel *et al.* 2003).

Figure 4 plots the localization error over time for the three different representations. The error was determined as the weighted average distance from the poses of the particles to the ground truth, where each weight is given by the importance factor of the corresponding particle. In the beginning of this experiment, the robot traveled through static areas so that all localization methods performed equally well. Close to the end, the robot traveled through the dynamic areas, which results in high pose errors for both alternative approaches. In contrast to that, our technique constantly yields a high localization accuracy and correctly tracks the robot.

To further illustrate, how our extended MCL is able to estimate the current state of the environment, Figure 5 plots the posterior probabilities for two different patches belonging to one sub-map. At time step 15, the robot entered the corresponding sub-map. At this point in time, the robot correctly identified, that the particles, which localize the robot in patch 1, performed much better than the samples using patch 0. Due to the resamplings in MCL, particles with a low importance weight are more likely to be replaced by particles with a high importance weight. Over a sequence
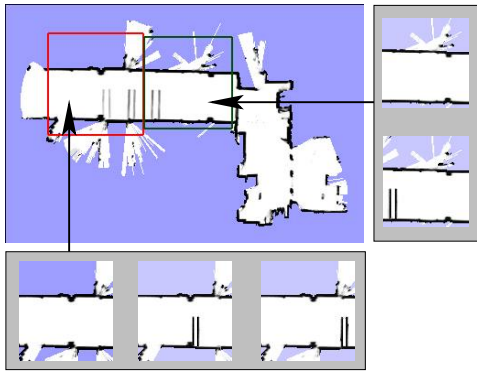
Figure 3: A patch-map with the different configurations for the individual patches.
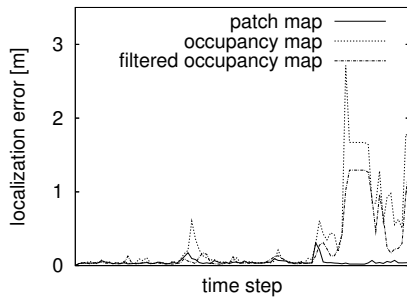

Figure 4: The error in the pose estimate over time. As can be seen, using our approach the quality of the localization is higher compared to approaches using occupancy grid maps.

of integrated measurements and resamplings, this led to an probability close to 1 that the environment looked like the map represented by patch 1 (which exactly corresponded to the ground truth in that situation).

Global Localization

Additionally, we evaluated all three techniques in a simulated global localization task. We compared our approach using two patches to represent the state of the door with standard MCL using occupancy grid maps (see Figure 6 and 7). In one experiment, the occupancy grid map contained the closed door and in the second one the open door. During localization, the robot mostly moved in front of the door, which was closed in the beginning and opened in the second phase of the experiment.

As can be seen in left column of Figure 6 and 7, the MCL approach which uses the occupancy grid that models the closed door as well as our approach lead to a correct pose estimate. In contrast to that, the occupancy grid, which models the open door causes the filter to diverge. In the second phase of the experiment, the door was opened and the robot again moved some meters in front of the door (see right column of the same figure). At this point in time, the MCL technique using the occupancy grid, which models the closed door cannot track the correct pose anymore, whereas our approach is able to correctly estimate the pose of the robot. This simulated experiment again illustrates that the knowledge about possible configurations of the environ-
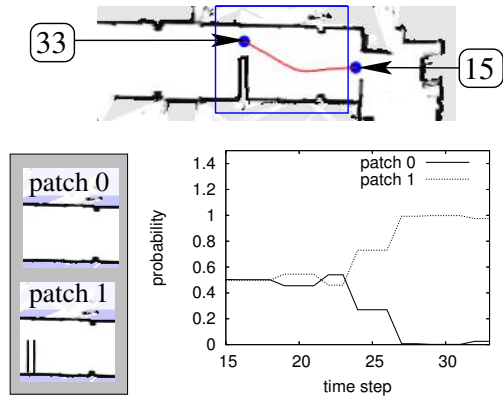

Figure 5: The image in the first row illustrates the traveled path with time labels. The left images in the second row depict the two patches and the graph plots the probability of both patches according to the sample set. As can be seen, the robot identified that patch 1 correctly models the configuration of the environment.

ment is important for mobile robot localization. Without this knowledge, the robot is not able to correctly estimate its pose in non-static environments.

Map Clustering

The last experiment is designed to illustrate the map clustering process. The input to the clustering was a set of 17 local grid maps. The fuzzy k-means clustering algorithm started with a single cluster, which is given by the mean computed over all 17 maps. The result is depicted in the first row of Figure 8. The algorithm then increased the number of clusters and re-computed the means in each step. In the fifth iteration the newly created cluster is more or less equal to cluster 3. Therefore, the BIC decreased and the clustering algorithm terminated with the model depicted in the forth row of Figure 8.

Conclusion

In this paper, we presented a novel approach to model quasi-static environments using a mobile robot. In areas where dynamic aspects are detected, our approach creates local maps and estimates for each sub-map clusters of possible configurations of the corresponding space in the environment. Furthermore, we described how to extend Monte-Carlo localization to utilize the information about the different possible environmental states while localizing a vehicle. Our approach as been implemented and tested on real robots as well as in simulation. The experiments demonstrate, that our technique yields a higher localization accuracy compared to Monte-Carlo localization based on standard occupancy grids as well as grid maps obtained after filtering out measurements reflected by dynamic objects.

One possibility to extend the presented approach is to combine our map model with techniques for simultaneous localization and mapping. Additionally, it would be interesting to apply techniques for online clustering. A further aspect, which has not been analyzed in detail is the
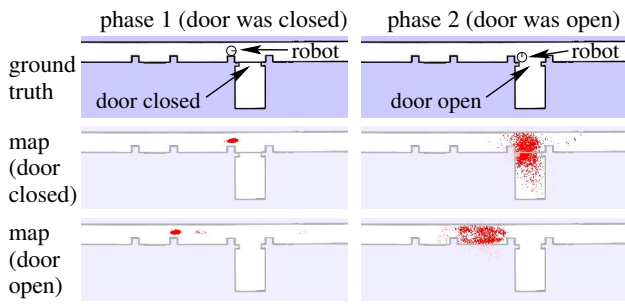
Figure 6: In the beginning the door was closed (left column) but was later on opened (right column). The first row depicts the ground truth, whereas the second row illustrates the particle distributions in case the door is supposed to be closed in the occupancy grid map, whereas no door was mapped in the third row.
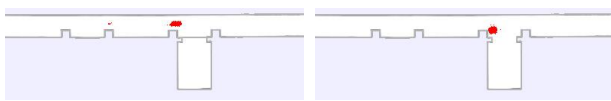

Figure 7: Particle clouds obtained with our algorithm for the same situations as depicted in Figure 6.

usage of topological information for dividing the environment into sub-maps. Such a segmentation would probably lead to more intuitive set of sub-maps. Nevertheless, the results obtained with our current implementation are more than promising. We belief that our representation can appropriately model most types of real world environments.

Acknowledgment

References

Anguelov, D.; Biswas, R.; Koller, D.; Limketkai, B.; Sanner, S.; and Thrun, S. 2002. Learning hierarchical object maps of non-stationary environments with mobile robots. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*.

Biber, P., and Duckett, T. 2005. Dynamic maps for long-term operation of mobile service robots. In *Proc. of Robotics: Science and Systems (RSS)*. To appear.

Dellaert, F.; Fox, D.; Burgard, W.; and Thrun, S. 1998. Monte carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.

Duda, R.; Hart, P.; and Stork, D. 2001. *Pattern Classification*. Wiley-Interscience.

Fox, D.; Burgard, W.; and Thrun, S. 1999. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research (JAIR)* 11.

Hähnel, D.; Triebel, R.; Burgard, W.; and Thrun, S. 2003. Map building with mobile robots in dynamic environments. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.
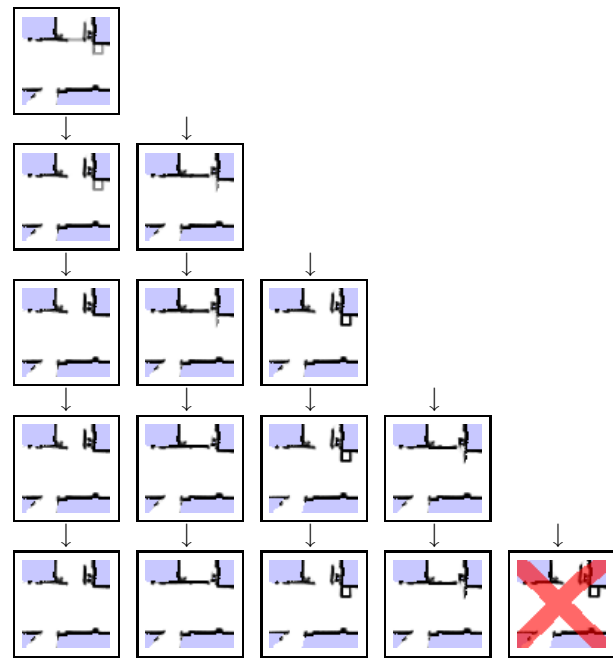
Figure 8: Iterations of the map clustering process. Our approach repeatedly adds new clusters until no improvement is achieved by introducing new clusters (with respect to the BIC). Here, the algorithm ended up with 4 clusters, since cluster 3 and 5 are redundant.

Hähnel, D.; Schulz, D.; and Burgard, W. 2002. Map building with mobile robots in populated environments. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

Montemerlo, M., and Thrun, S. 2002. Conditional particle filters for simultaneous mobile robot localization and people tracking. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.

Montemerlo, M.; Thrun, S.; Koller, D.; and Wegbreit, B. 2002. FastSLAM: A factored solution to simultaneous localization and mapping. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*.

Moravec, H., and Elfes, A. 1985. High resolution maps from wide angle sonar. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.

Murphy, K. 1999. Bayesian map learning in dynamic environments. In *Neural Info. Proc. Systems (NIPS)*.

Romero, L.; Morales, E.; and Sucar, E. 2001. A hybrid approach to solve the global localozation problem for indoor mobile robots considering sensor's perceptual limitations. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*.

Schwarz, G. 1978. Estimating the dimension of a model. *The Annals of Statistics* 6(2).

Wang, C.-C., and Thorpe, C. 2002. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.