Online Map Recognition using Bayesian Updates

Fengkai Chen, Jake Olkin, Cameron Kisailus, Tianyang Shi, Sihang Wei

I. Introduction

In order to reliably deploy robots in large-scale environments, it is fundamental for robots to have accurate localization. Robot relocalization is the act of determining the position in a robot in a previously mapped environment. Relocalization in small-scale environments is usually done by maintaining a contiguous map of the entire environment ([4], [15]), but this does not scale. Methods to combat this scaling problem have proposed splitting large, global maps into smaller metrical maps [7], [11], [12]. However, this adds another dimension to the relocalization problem, since the algorithm must additionally determine which map is it located on. Probabilistic localization algorithms are governed by the total law of probability, so it is impossible to indicate low, or even zero, belief a robot is anywhere on a given map (i.e. a particle filter will always maintain some belief of the robot location).

To this end, we present a system for online map recognition method using Bayesian methods. We model the map recognition problem as a Hidden Markov Model where the state is the belief that we are in a given map and observations are particle distributions. We developed a heuristic-based likelihood model to model the conditional probability of a particle distribution given a map. We present experimental results show our algorithm correctly matches the maps to the environments.

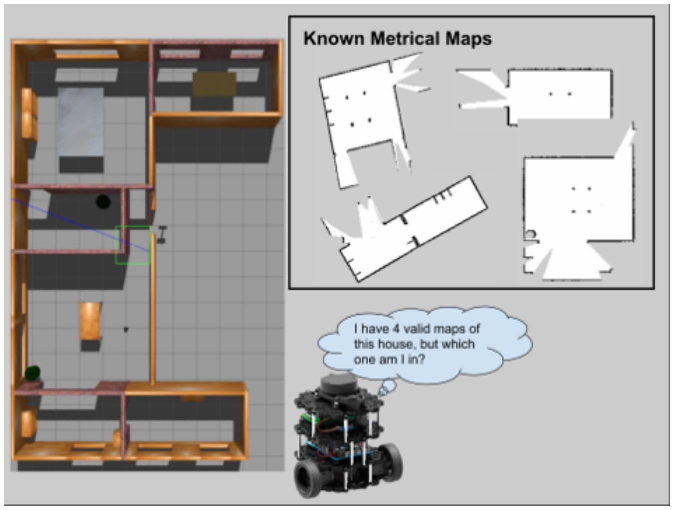


Fig. 1. Visual example of the problem statement

II. Related Works

A. Relocalization

Previous attempts to solve the relocalization problem assume that there is only a single map being considered. However, there are some similarities between relocalization across multiple maps and localization across large global maps, or topological maps.

In [14], relocalization is performed over a topological map. While the nature of a topological map contains much less metric information than the occupancy grids our method assumes we have access to, the nature of choosing which node in the topological map the robot is occupying is analogous to choosing the correct metrical map for relocalization. However, this method does heavily differentiate from ours by using image input as opposed to LIDAR point clouds. By using LIDAR point clouds, we can take better advantage of our occupancy grid maps.

An addition to the Atlas SLAM framework ([1]) localizes on a global map built by applying scan-matching algorithm on multiple local maps. The similar aspect of this work to our method is the nature of matching the local maps to their global maps to expand their global map. This work builds each local map individually, and then performs matching from the local maps to the global map, which is a major departure from our work since we do not perform additional mapping. We instead attempt to relocalize directly from the particle distributions.

ScanContext ([8]) is another example of place recognition from point clouds. The work uses spatial descriptors to generate similarity scores between places mapped. Similar to our method, this work does not rely on any previous training or histogram data. This is largely optimized to be used for loop detection, so instead of taking multiple maps to analyze, this method splits a single map into multiple different parts to determine the different places to analyze.

Additionally, there are a number of works that use deep learning to perform place recognition using images ([2], [3]) or other sensors like radar ([10]). While these methods have found some success, our method does not require any form of training, which means it can operate in environments regardless of available training data.

B. AMCL

Adaptive Monte-Carlo Localization (AMCL) [5] is a particle filter based relocalization approach. Unlike traditional particle filters, AMCL adapts the number of particles to maintain a larger number of particles when the robot's location is uncertain and a lesser number when the certainty is high. AMCL has been proven to be more effective in global relocalization than traditional (constant sample size) particle filters or likelihood-based adaptive particle filters as in [6] and [9]. AMCL calculates the number of particles necessary to remain within some ε of the true distribution through a method they call KLD-sampling. Because of its performance in the global relocalization task as well as the pre-compiled ROS packages, we chose to use AMCL in our experiments.

C. Hidden Markov Model

A Hidden Markov Model (HMM) [13] is a probabilistic graphical model used to model the joint distribution of a state random variable, X_t, observed through noisy measurements, Y_t. HMM have two major components: transition model, $P(x_t \mid x_{t-1})$ and likelihood model, $P(y_t \mid x_t)$. The joint distribution of a HMM is defined as:

$$P(X,Y) = P(x_0) \prod_{t=1}^{T-1} P(x_t \mid x_{t-1}) P(y_t \mid x_t) \qquad (1)$$

III. Problem Statement

Suppose an autonomous agent has generated multiple, disconnected metrical maps of an environment. The robot is then set to localize itself to this set of maps without an initial estimate of its prior position. We seek to determine which map is most likely for the robot to be currently located. Formally, given a set of metrical maps M, and a stream of particle distributions for each map P_m, we model the problem as a HMM. Our state, X_t, represents our belief we are in a given map. Our observations, Y_t are the particle distribution $P_m \; \forall \; m \in M$. We assume a static state transition model and implemented a heuristic based likelihood model explained in IV-B. Figure 2 shows a graphical representation of our HMM. We utilize 1 to infer the belief that we are in a given map.

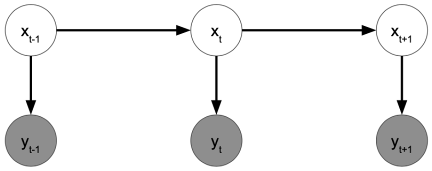


Fig. 2. Hidden Markov Model(HMM): Dark nodes are observed, light nodes are hidden.

For our particular formulation, we are using occupancy grids to represent mapped environments and particle distributions generated by AMCL as observations. Since our likelihood model operates on particle distributions, our algorithm is sensor agnostic.

IV. Methodology

A. Map Recognition Pipeline

Given a large-scale map, we will separate it into a set of small-scale maps and perform accurate localization on them. Because the current position of robot is unknown, for each sub-environment, we take the LIDAR scans as sensor data and localize the robot over each small-scale map using AMCL. We then update the belief of being located in that map according to Equation 1. The implementation of our algorithm is shown in Algorithm 1.

Algorithm 1 Map Recognition Pipeline

```
Require: LIDAR scans of Environment $Z \leftarrow \{z_1, z_2, ...z_N\}$, Timesteps $T$, Number of maps N, Maps $M \leftarrow \{m_1, m_2, ...m_N\}$
  $X_i^0 \leftarrow 1/N \; \forall \; i \in [1, 2, ..., N]$    ▷ Initialize with uniform belief in all maps
  for $t = 1 \rightarrow T$ do
      for $i = 1 \rightarrow N$ do
          $P_i \leftarrow AMCL(M_i, Z_t)$    ▷ Generate particle distribution
          $X_i^t \leftarrow X_i^{t-1} * Likelihood(P_i \mid M_i)$    ▷ Update belief
      end for
  end for
  $m^* = \arg\max_i X_i^T$    ▷ Get map with max confidence
  return $m^*$
```

B. Belief Update Equation

We use two main factors in calculating our place recognition scores from the adaptive particle filter output: covariance of the estimated position, and the ratio of particles that exist in free space. The first factor is an effective measurement as it provides the particle filter's own confidence in its estimate. The particle filter should only reach consensus if it can properly match the features of the map of the environment to the laser scans it is given. If there is no strong match there, then the particle filter will not converge, resulting in a high covariance.

However, there are some cases where the particles may still reach some convergence on incorrect maps, especially if futures across maps are similar. To mitigate the intensity of this false belief, we also factor in how many particles are not in free space of the given map. Particles that occupy occupied or unknown space are physically impossible locations for a robot. Therefore, a particle distribution with a high number (ratio) of physically impossible robot locations decreases the likelihood that we are located on that map. Figure 3 demonstrates the necessity of this metric by contrasting the position of particles on a map that matches the environment a robot is currently occupying and a map that does not match the current environment. Although both particle filters show some sign of convergence, the distribution on the map in the right is almost entirely in unknown space.

We combine these two factors in equation 2, where Σ_i is the covariance of particle filter pose estimate for map i, $f(p_i, m_i)$ is the percentage of particles not occupying free space. We designed this belief update function to reward maps that have both low covariance, and have the most particles in free space positions.

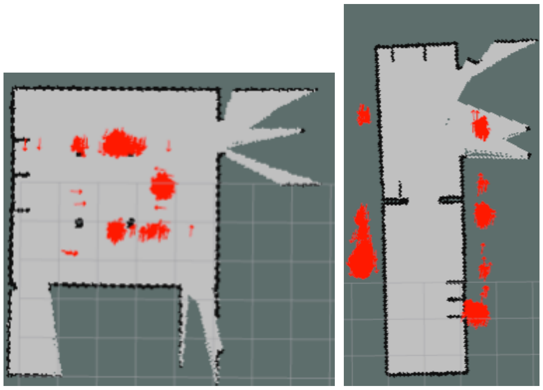


Fig. 3. Both images show the spread of particles form a particle filter after 60 seconds attempting to localize. The map on the left matches the environment the robot is currently receiving laser scans from, whereas the map on the right is a different candidate map that it could have been occupying.

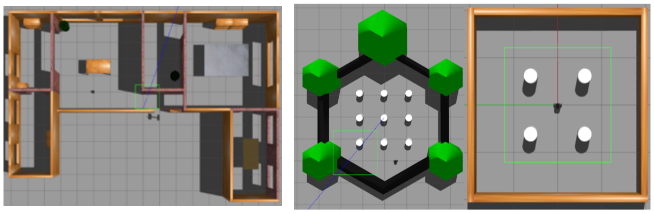
$$P(m_i|p_i,\Sigma_i) \propto \frac{1}{f(p_i,m_i)||\Sigma_i||} \tag{2}$$

Since we maintain belief over each map accumulated over previous laser scans, we update our belief by multiplying the our belief for the current time step by our accumulated belief, and then normalizing.

V. Experiments and Results

A. Data collection

All data in this work is collected by ourselves. We conduct the experiments with Gazebo simulation and visualize the localization process with RVIZ. We used a ROBOTIS Turtlebot3 as our robot platform. As shown in Figure 5, we chose 6 environments, including 4 rooms from the virtual apartment and 2 virtual worlds of different shapes. After mapping each environment, we conducted our experiment by attempting to localize in each map given a 3 meter X 1 meter @ 0.1 m/s robot trajectory. During each trial we record the particle distributions, estimated pose and covariance at each filter update.

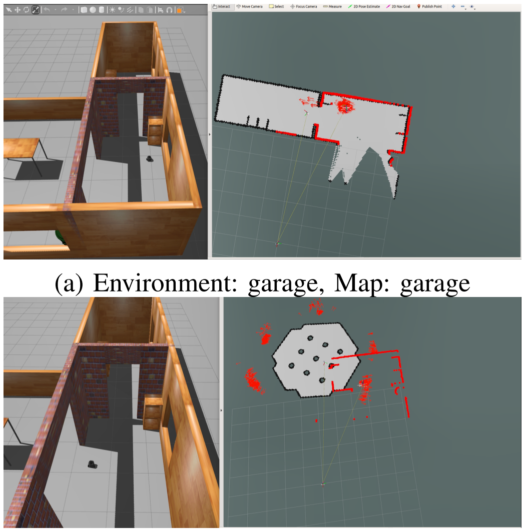


(a) Virtual apartment

(b) Virtual Worlds

Fig. 5. Experiment environments

For example, Figure 6 shows particle distributions during 2 different trials. In both trials, we are using LIDAR scans coming from the $garage$ environment but attempting to localize in the $garage$ and $turtle_world$ in A and B, respectively. When localizing on the matched map, the particles converge quickly and most of them are located in the free space of the map (white area in RVIZ). As contrast, when localizing on a incorrect map, the particles are distributed in a large range with most of them occupying invalid cells of the map (gray and black area in RVIZ). This performance demonstrates that our approach is feasible. We could find the target map by analyzing the covariance and the rate of invalid particles from AMCL data.



(a) Environment: garage, Map: garage

(b) Environment: garage, Map: turtle world

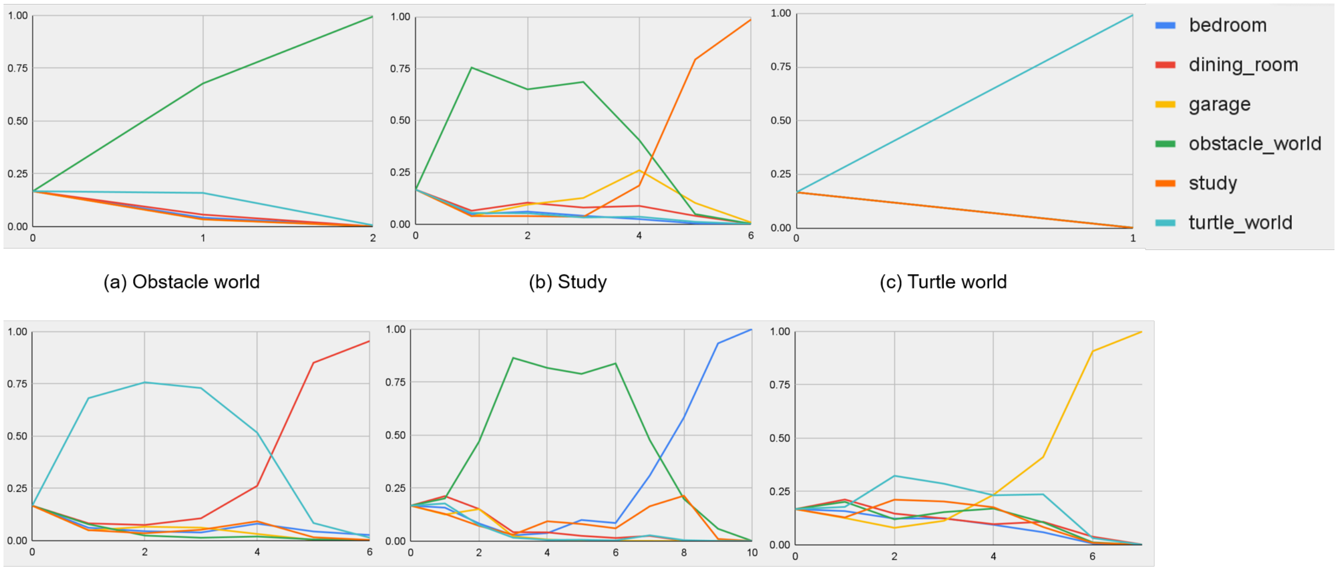
Fig. 6. Particle distributions during localization on different maps

B. Final Belief Results

During each experiment, we would begin with a uniform belief distribution over the candidate maps, and then update this belief after each particle filter update. The particle filter does not update at a constant rate over time, instead only updating after the robot has moved a minimum distance, or rotated a minimum amount. We set the particle filter parameters to update once every 20cm of translation, and every 30° of rotation. Since our method relies on output from the particle filter, we define that the k time step of our method refers to our output after k particle filter updates.

We also set a maximum belief threshold to stop performing belief updates once we were confident we knew which environment the robot was traversing. Once any of our beliefs exceeded 0.95, we would cease updates and take the rankings after that time step as our final rankings.

You can see our experimental results in Table I, which displays the final rankings from each experiment. We identified the correct environment in all of our experiments. Additionally, in Table II, we show how many time steps it took until our algorithm finished. We used at most 10 time steps to identify our current environment. We depict the evolution of our belief over time in each experiment in Figure 4. There are times where our algorithm does hold false beliefs, which tend to occur when one of the incorrect maps false converges it's position estimate, resulting in a low covariance. However, this is offset by the number of particles not occupying valid locations, and never breaches our 0.95



(a) Obstacle world

(b) Study

(c) Turtle world

(d) Dining room

(e) Bedroom

(f) Garage

Fig. 4. Evolution of our map belief over time. The X axis represents the the number of time steps (which is equal to the number of particle filter updates that are performed), and the Y axis represents the belief of each map.

		Environment					
		Bedroom	**Study**	**Garage**	**Dining Room**	**Turtle World**	**Obstacle World**
Map	**Bedroom**	**1**	6	6	2	4	4
	Study	3	**1**	4	4	5	5
	Garage	6	2	**1**	6	6	6
	Dining Room	4	3	2	**1**	2	3
	Turtle World	5	4	3	5	**1**	2
	Obstacle World	2	5	5	3	3	**1**

TABLE I

MAP RANKINGS FROM EXPERIMENTATION: EACH COLUMN REPRESENTS THE ENVIRONMENT THE ROBOT COLLECTED LASER DATA FROM, AND EACH NUMBER IN A COLUMN IS THE RANKING OF THE BELIEF OF THE CORRESPONDING MAP.

Environment					
Bedroom	Study	Garage	Dining Room	Turtle World	Obstacle World
10	6	7	6	1	1

TABLE II

NUMBER OF STEPS TAKEN BY EACH ENVIRONMENT TO REACH 0.95 BELIEF

belief threshold. Over time, the correct map always surpasses any false beliefs.

C. Belief Update Ablation Study

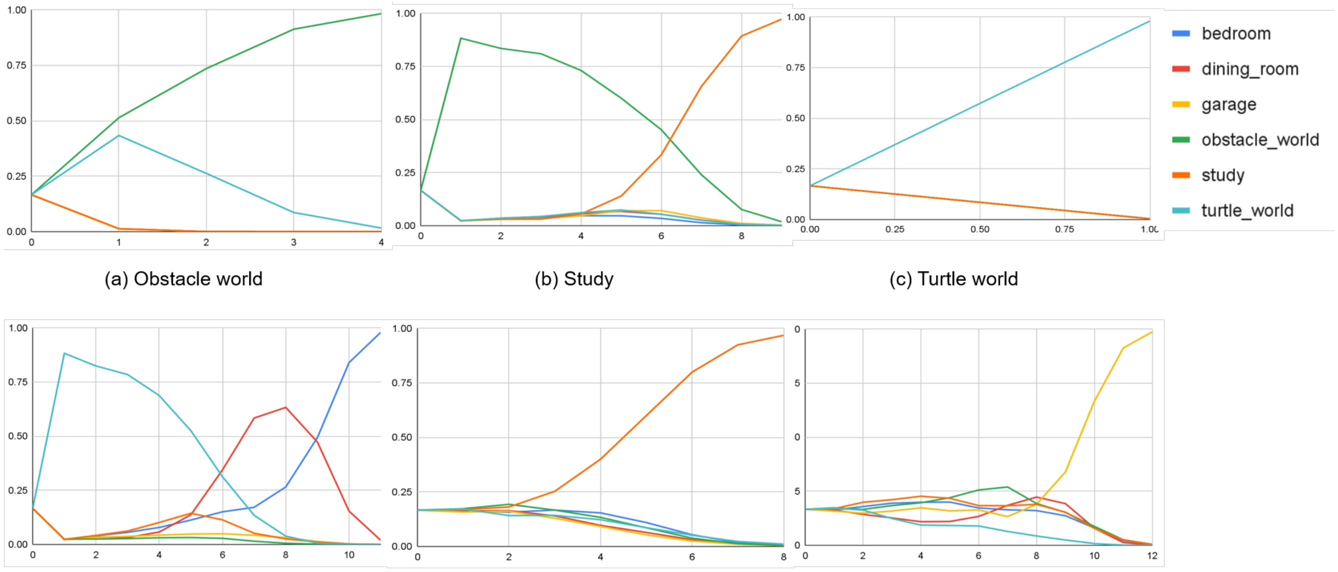
To see how the different factors of our belief update function contributed to our final ranking, we conducted ablation experiments, using the two following variants of our belief update function:

$$P(m_i|p_i, \Sigma_i) \propto \frac{1}{||\Sigma_i||}$$

$$P(m_i|p_i, \Sigma_i) \propto \frac{1}{f(p_i, m_i)}$$

where Σ_i is the covariance of particle filter pose estimate for map i, $f(p_i, m_i)$ is the percentage of particles not occupying free space. The first equation only uses the covariance of the pose estimate, whereas the second only uses the number of particles in overlapping space. In the ablation experiments, we used the same 0.95 threshold, and the same parameters regarding updates to the particle filter. The evolution of our belief using only covariance can be seen in Figure 7, while the evolution of our belief using only overlapping particles can be seen in Figure 8.

Some environments, such as the $garage$ and $turtleworld$, match correctly when using both the covariance and the overlap. However, there are numerous examples where the incorrect map will score highly on one metric, but not the



(a) Obstacle world

(b) Study

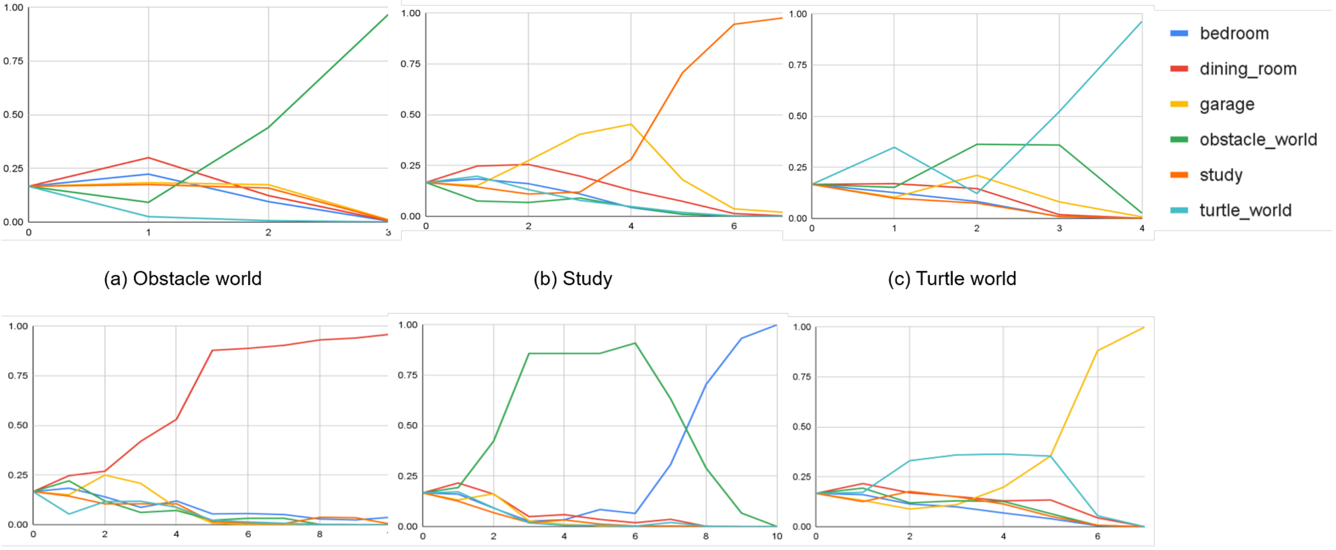
(c) Turtle world

(d) Dining room

(e) Bedroom

(f) Garage

Fig. 7. Map belief updates - considering the particles covariance only. The X axis represents the the number of time steps (which is equal to the number of particle filter updates that are performed), and the Y axis represents the belief of each map



(a) Obstacle world

(b) Study

(c) Turtle world

(d) Dining room

(e) Bedroom

(f) Garage

Fig. 8. Map belief updates - considering the invalid particle rate only. The X axis represents the the number of time steps (which is equal to the number of particle filter updates that are performed), and the Y axis represents the belief of each map

other. The most extreme example of this is the $dining_room$ environment using only covariance. In this case, by the end of our experiment, the bedroom was what we believed was our true environment. This demonstrates why we need to use the combination of the two criteria together, since they compliment each other so we do not falsely believe we are occupying the wrong map.

VI. Conclusion and Future Work

In this work, we work toward online map recognition by modeling the problem as a Hidden Markov Model and using Bayesian belief updates. We also developed a heuristic-based likelihood model. In experiments, we show our method correctly recognizes the map 100% of the time. Our experiments also display a relatively quick time to 95% belief, especially in very distinct environments. Furthermore, our work can be applied to much more complex scenarios, as long as with sufficient maps and sensor data to perform particle filter.

From our results, neither the covariance and invalid particles can give us a accurate prediction solely. Similar environments and landmarks made the particles filter to converge quickly, which will return a reasonable covariance and confused the belief update. By adding invalid filter as a factor, we could penalize the particles at occupied space or unknown space even if they converge at a rapid rate.

While our experiments demonstrate a promising result in Gazebo simulation, its performance on large datasets still remains unknown. In the future, we shall perform map recognition on more different environments. In addition, we will examine and optimize the performance of our work on similar maps. It is also interesting to combine sensor information as a factor in belief update process.

The dataset and code of this project can be found at https://github.com/camkisailus/ROB530_FinalProject.We also uploaded a video presentation at https://www.youtube.com/watch?v=pzFOLR6G_VEt=7s.

References

[1] Michael Bosse and Robert Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, 27(6):667–691, 2008.

[2] Zetao Chen, Adam Jacobson, Niko Sünderhauf, Ben Upcroft, Lingqiao Liu, Chunhua Shen, Ian Reid, and Michael Milford. Deep learning features at scale for visual place recognition. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3223–3230, 2017.

[3] Zetao Chen, Obadiah Lam, Adam Jacobson, and Michael Milford. Convolutional neural network-based place recognition. *CoRR*, abs/1411.1509, 2014.

[4] Dieter Fox. Kld-sampling: Adaptive particle filters. *Advances in neural information processing systems*, 14, 2001.

[5] Dieter Fox. Adapting the sample size in particle filters through kld-sampling. *The international Journal of robotics research*, 22(12):985–1003, 2003.

[6] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349):2–2, 1999.

[7] Naveed Islam, Khalid Haseeb, Ahmad Almogren, Ikram Ud Din, Mohsen Guizani, and Ayman Altameem. A framework for topological based map building: A solution to autonomous robot navigation in smart cities. *Future Generation Computer Systems*, 111:644–653, 2020.

[8] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809, 2018.

[9] Daphne Koller and Raya Fratkina. Using learning for approximation in stochastic processes. In *ICML*, pages 287–295, 1998.

[10] Jacek Komorowski, Monika Wysoczanska, and Tomasz Trzcinski. Minkloc++: Lidar and monocular image fusion for place recognition. *CoRR*, abs/2104.05327, 2021.

[11] Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 4845–4851. IEEE, 2004.

[12] Joseph Modayil, Patrick Beeson, and Benjamin Kuipers. Using the topological skeleton for scalable global metrical map-building. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 2, pages 1530–1536. IEEE, 2004.

[13] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[14] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 2, pages 1023–1029 vol.2, 2000.

[15] Shaohua Kevin Zhou, Rama Chellappa, and Baback Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Transactions on Image Processing*, 13(11):1491–1506, 2004.