2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10611256

Robot Navigation in Unseen Environments using Coarse Maps

Chengguang Xu, Christopher Amato, Lawson L.S. Wong

Abstract— **Metric occupancy maps are widely used in autonomous robot navigation systems. However, when a robot is deployed in an unseen environment, building an accurate metric map is time-consuming. *Can an autonomous robot directly navigate in previously unseen environments using coarse maps?* In this work, we propose the Coarse Map Navigator (CMN), a navigation framework that can perform robot navigation in unseen environments using different coarse maps. To do so, CMN addresses two challenges: (1) novel and realistic visual observations; (2) error and misalignment on coarse maps. To tackle novel visual observations in unseen environments, CMN learns a deep perception model that maps the visual input from various pixel spaces to the local occupancy grid space. To tackle the error and misalignment on coarse maps, CMN extends the Bayesian filter and maintains a belief *directly* on coarse maps using the predicted local occupancy grids as observations. Using the latest belief, CMN extracts a global heuristic vector that guides the planner to find a local navigation action. Empirical results demonstrate that CMN achieves high navigation success rates in unseen environments, significantly outperforming baselines, and is robust to different coarse maps.**

I. INTRODUCTION

Autonomous mobile robots have made substantial contributions to modern society in the industrial, service, and medical fields [1]. On these platforms, the navigation system plays a particularly important role because it grants mobile robots the ability to move toward specified goals [2].

In robotics, a typical navigation system requires a global metric map (e.g., occupancy map [3]) and follows a classic pipeline that consists of perception, position estimation, and path planning [4]. Although such a navigation system is robust and effective, it requires an accurate map beforehand. For example, a robot vacuum in Figure 1 needs to build an accurate occupancy map before it can clean the room. However, building an accurate map is time-consuming and possibly difficult (e.g., in toxic factories or underground tunnels [5], [6]). Mapping algorithms may also result in local minima and errors in the maps, which require expertise to analyze and correct [5]. Furthermore, accurate maps need to be frequently updated in dynamic environments.

In this work, we propose the Coarse Map Navigator (CMN), which navigates without an *a priori* metric map. Instead, CMN only requires a coarse representation of the environment to be provided. The *coarse-grid* and *hand-drawn* maps shown in Figure 1 are two examples of coarse maps. Such coarse maps are much easier for non-experts to provide.

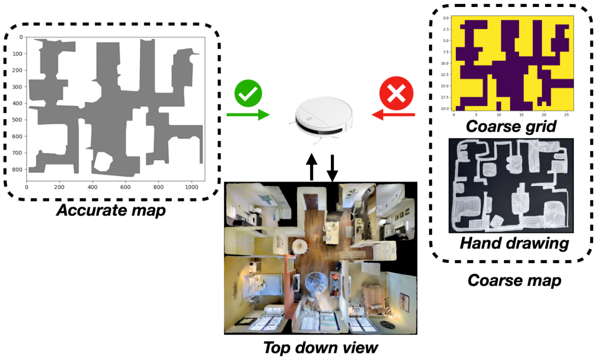

Fig. 1. A robot vacuum needs to build a global occupancy map before it performs the cleaning service. However, it is unable to use coarse representations (e.g., *coarse-grid* maps or *hand-drawn* maps) of the environments. In this work, we develop CMN, which can use different coarse maps for robot navigation. Note that the size of a *coarse-grid* map is much smaller than that of a *metric* map (i.e., 20×25 vs. 900×1000).

However, since the coarse maps are rough representations, they may contain local errors and global misalignment with the real world.

Using coarse maps, CMN follows the classic navigation pipeline with extra modifications. As shown in Figure 2, given a coarse map of an unseen environment, CMN takes in visual inputs and outputs a reactive action. To tackle novel visual observations in unseen environments, CMN learns a local occupancy predictor using deep neural networks [7], mapping visual observations to predicted local occupancy grids. Such a perception model makes CMN generalize well to novel observations in unseen environments. Using the predicted local occupancy grids as observations, CMN makes a simple yet effective modification to the discrete Bayesian filter [8] and maintains a belief of the robot's location *directly* on the coarse map. The modification makes the filter robust to errors and misalignment caused by coarse maps. Given the latest belief from the Bayesian filter, CMN computes a global heuristic vector using a grid search algorithm (e.g., A^* search). Finally, CMN develops a tree-structured planner to find a local action that moves the robot in a similar direction indicated by the global heuristic.

We evaluate the performance of CMN in Habitat [9], a photo-realistic domain for robot navigation. We design systematic experiments using two types of maps (*coarse-grid* and *hand-drawn* maps) in three tasks with varying challenges. Compared with the classic navigation pipeline that always requires a globally accurate metric map, empirical results demonstrate the effectiveness and robustness of CMN, improving the average navigation success rate by $20\% - 38\%$ using different *coarse-grid* maps and by 17% using *hand-drawn* maps in previously unseen environments.

This work was supported by NSF Grants #2024790 and #2142519, and by ARO Grant W911NF20-1-0265.

Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA {xu.cheng, c.amato}@northeastern.edu ; lsw@ccs.neu.edu

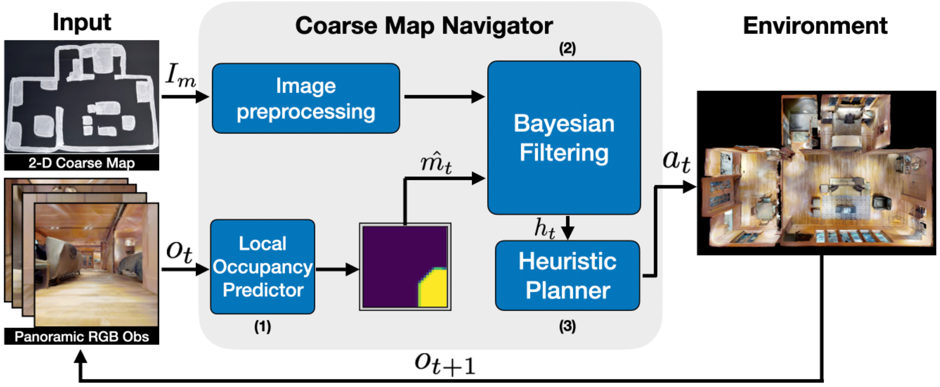

Fig. 2. Framework overview. CMN requires a 2-D coarse map I_m and a panoramic RGB observation o_t as input. (1) The local occupancy predictor predicts the top-down local occupancy grid $\hat{m}_t \in \hat{\Omega}$ from the panoramic RGB observation (Section IV-B). (2) Using the predicted local occupancy grid as the observation, CMN estimates the robot's location on the coarse map using a modified Bayesian filter and outputs a global heuristic vector h_t (Section IV-C). (3) Finally, CMN plans a local reactive action a_t based on the global heuristic vector h_t (Section IV-D).

II. RELATED WORK

Using a global map for robot navigation has been well-studied in the literature. The global map is a model of the environment that varies from a complete CAD model to a simple graph of interconnections or interrelationships between the objects in the environment [10]. Among the different map representations, the grid representation [3], [11], usually known as "occupancy maps", is widely used in robot localization [12], [13], [14], [15] and navigation [16], [17]. However, they usually assume the maps are globally accurate and metrically consistent with the environment. Unlike these methods, CMN uses coarse maps (e.g., *coarse-grid* and *hand-drawn* maps) that might contain metric inconsistencies with the environment.

There is much less work on robot navigation without a metrically consistent map. [18], [19], [20] use topological representations of the environments, which could be extracted from provided blueprints or computed from a SLAM map [21]. However, these topological methods typically still rely on metric sub-maps for local navigation, which are collected prior to construction of the topological map. Thus topological navigation methods cannot be easily adapted to navigate in new environments given only the graph. In contrast, CMN navigates using the graph alone, without needing to first create a metric map of the environment.

Hand-drawn maps have also been used in robot navigation. However, [6] points out that most of the research [22], [23], [24], [25], [26] focuses on designing human-robot interaction systems where *hand-drawn* sketches are used to generate human-robot communication signals. Other research such as [27] aims to evaluate the navigability of *hand-drawn* maps using multiple-hypotheses tracking. [28] studies *hand-drawn* map interpretation and matching, which still converts the map into a topological representation. [29] proposes an algorithm to determine the relationship of the objects between a *hand-drawn* sketch map and the occupancy grid built by a robot to facilitate human-robot interaction during navigation. In summary, those systems involve human operators in-the-loop, whereas CMN is a fully autonomous navigation system that can use *hand-drawn* maps.

Fully autonomous navigation systems using *hand-drawn* maps are studied in [6], [30], [31], [32]. [30] proposes an approach to fit the *hand-drawn* maps to the local occupancy obtained from the stereo sensor using the FastSLAM algorithm [33] with particle swarm optimization. Most closely related to our work is [6] (extending [5] from localization to navigation), which proposed fitting the local occupancy to the *hand-drawn* map. However, the above work assume having the initial robot location to track the robot's location. For CMN, we consider a more challenging setup in which the initial robot location is unknown. Instead, CMN has to estimate the robot's location from a uniform distribution. Additionally, [6] assumes the existence of a diffeomorphism between the sketch map and the world. To estimate this, [6] requires the robot's location as input. However, we assume no robot location information is available to CMN.

III. PROBLEM STATEMENT

The problem of visual navigation in unseen environments can be formulated as a partially observable Markov decision process (POMDP) [34]. In particular, we call it the *env-POMDP*. In our navigation setup, we assume that the robot is additionally provided with a 2-D coarse map I_m. Therefore, in addition to the *env-POMDP*, we formulate a related POMDP, the *map-POMDP*, that operates on the corresponding 2-D coarse map, with much smaller observation and state spaces. We argue that the *map-POMDP* can be used to solve the navigation task without explicitly solving the original *env-POMDP*, which motivates our use of coarse maps.

The *env-POMDP*, defined in the real world, is a POMDP, represented as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega, \gamma \rangle$. $\mathcal{S}$, $\mathcal{A}$, Ω are finite sets of states, actions, and observations, respectively. $\mathcal{T}$: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition function. $\mathcal{O}$: $\mathcal{S} \times \mathcal{A} \times \mathcal{O} \rightarrow [0, 1]$ is the observation probability. $\mathcal{R}$: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is the goal-conditioned reward function. γ is the discount factor. In visual navigation, the states s consist of the 2-D location and the orientation (x, y, θ). The observations o are egocentric images captured at corresponding states s. The actions a are also egocentric movements.

Similarly, we also define the *map-POMDP* for the coarse map, represented as $\langle \hat{\mathcal{S}}, \hat{\mathcal{A}}, \hat{\mathcal{T}}, \hat{\mathcal{R}}, \hat{\mathcal{O}}, \hat{\mathcal{Z}}, \gamma \rangle$. We assume that the *map-POMDP* is related to the *env-POMDP* but has two mismatches. First, the observation space ($\hat{\mathcal{Z}} \neq \mathcal{Z}$) and likelihood ($\hat{\mathcal{O}} \neq \mathcal{O}$) are different, resulting in *perception mismatch* because the observation z in the unseen environment is egocentric images, while the observation $\hat{z}$ on the map is top-down local occupancy. Second, states on the coarse map do not necessarily correspond to states in the unseen environment, resulting in *motion mismatch*. For example, moving one state on the map might correspond to traversing multiple states in the environment.

We assume the map is coarse such that $|\hat{\mathcal{S}}| \ll |\mathcal{S}|$ and $|\hat{\Omega}| \ll |\Omega|$. Therefore, instead of maintaining a belief $Bel(s_t)$ over $\mathcal{S}$, CMN maintains a belief $\widehat{Bel}(\hat{s}_t)$ over $\hat{\mathcal{S}}$. To do so, CMN learns a mapping function $\mathcal{F}_\psi : \Omega \rightarrow \hat{\Omega}$ that maps observations from images to local occupancy grids (Section IV-B). CMN also assumes a stochastic relationship between $\mathcal{A}$ and $\hat{\mathcal{A}}$ (Section IV-C) because the map coarseness is unknown. To ensure the belief $\widehat{Bel}(\hat{s}_t)$ is useful, we assume that there exists an unknown diffeomorphism between the map and the real world (Section IV-A), meaning that any arbitrary robot trajectory in the real world can be described on the coarse map [6]. Therefore, given the belief $\widehat{Bel}(\hat{s}_t)$, CMN derives a policy $\hat{\pi}(h_t|\widehat{Bel}(\hat{s}_t), \hat{s}_g)$ of the *map-POMDP* that is used as the heuristic to generate the policy $\pi(a_t|h_t, s_g)$ of the *env-POMDP* (Section IV-D), where h_t is the heuristic vector and $\hat{s}_g$ is the goal state on the coarse map.

IV. COARSE MAP NAVIGATOR

CMN consists of three components: (1) a local occupancy predictor that maps visual observations (images) to local occupancy grids, (2) a modified Bayesian filter that maintains a belief over the robot's locations on the coarse map, and (3) a tree-structured heuristic planner that plans a reactive action in the environment based on the heuristic vector computed from the latest belief. We first explain the assumptions we make on coarse maps, before describing the CMN components.

A. Assumptions on coarse maps

Using CMN, the robot is provided with a coarse map of an unseen environment beforehand. In particular, we consider two types of coarse maps: *coarse-grid* and *hand-drawn* maps. Unlike other methods that use blueprints to generate topological representations [20], CMN uses the original maps and treat them as coarse occupancy grids.

We formally define the relationship between the real world and the coarse map based on the manifold formalism in [6]. A coarse map is modeled as a 2-D Euclidean space $E_m := (I_m, R_m)$, where I_m is a rasterized image and R_m is the reference frame. Similarly, the real world is also modeled a 2-D Euclidean space $E_w := (I_w, R_w)$. We assume that there exists a diffeomorphism $\mathcal{F}_d : I_w \in \mathbb{R}^2 \rightarrow I_m \in \mathbb{R}^2$ that transforms the pixels representing the free space between the two rasterized image representations. The formulation implies that any robot trajectory in the real world can be represented on the coarse map using the diffeomorphism operator $\mathcal{F}_d$. For the full formulation, please see [6], [5].

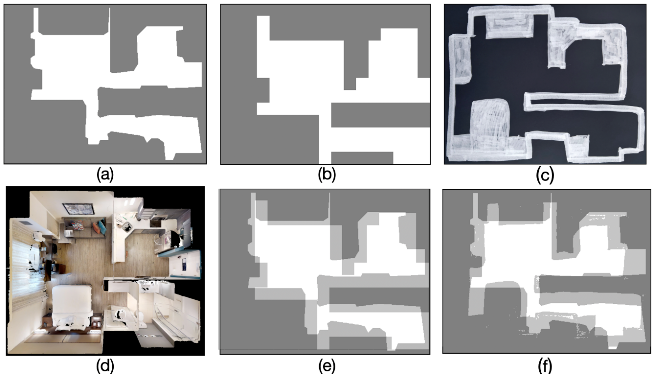

Fig. 3. Given one example environment in Gibson, (d) shows the top-down 3-D view. (a) - (c) show the *metric* map, the *coarse-grid* map, and the *hand-drawn* map, respectively. (e) shows the misalignment between the metric map and the coarse-grid maps. (f) shows the misalignment between the metric map and the hand-drawn map. The coarse maps and the metric maps are manually aligned. Dark gray represents the overlapped region, whereas light gray represents the mismatched region. There exists a significant difference between the coarse maps and the metric maps. (The color scheme is changed to highlight the map differences.)

We only use the formulation to define the map-environment relationship. However, unlike [6], we do not estimate the unknown diffeomorphism operator $\mathcal{F}_d$ because such estimation requires the robot's location, which we assume is unavailable to CMN. To obtain a *coarse-grid* map, we downsample the real environment into a coarse occupancy grid, where each grid cell is considered occupied if the occupancy rate is above a threshold. The *hand-drawn* maps are drawn by a human operator. Figure 3 shows an example of a *coarse-grid* map and a *hand-drawn* map.

B. Predicting the local occupancy from visual observations

Given a coarse map image I_m, CMN maintains a belief $\widehat{Bel}(\hat{s})$ over the states $\hat{\mathcal{S}}$ on it. Therefore, we first need to map the environment observation space Ω to the coarse map observation space $\hat{\Omega}$. Furthermore, it is also important that CMN can quickly generalize to novel observations in unseen environments. To this end, we train a deep perception model that predicts local occupancy grids from visual observations (Figure 4), which we call the local occupancy predictor.

As shown in Figure 2, the local occupancy predictor takes in a panoramic observation o_t consisting of 4 RGB images and outputs an $n \times n$ local occupancy grid $\hat{m}_t$. Formally, the local occupancy map predictor defines a mapping function $\mathcal{F}_\psi : \Omega \rightarrow \hat{\Omega}$. We model $\mathcal{F}_\psi$ as a deep neural network parameterized by ψ. Similar to [35], the local occupancy predictor adopts a deep convolutional autoencoder architecture and is extended to tackle panoramic image observations.

To train the local occupancy predictor, we built an offline dataset $\{(o_i, m_i)\}_{i=1}^N$ that contains pairwise training data, where o_i is the panoramic observation and m_i is the corresponding ground truth local occupancy. Using the pairwise data (o_i, m_i), we train the local occupancy predictor by minimizing the mean squared error between the predicted local occupancy $\hat{m}_i$ and the ground truth m_i using stochastic

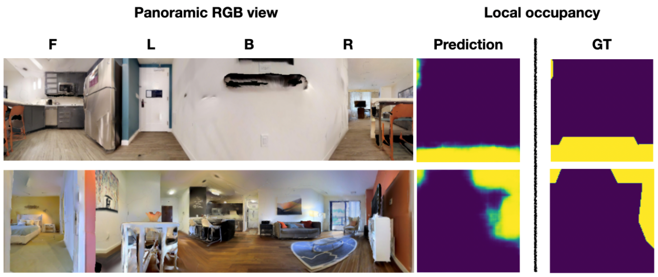

Fig. 4. Given a panoramic RGB observation, the deep perception model makes a good local occupancy prediction. F, L, B, and R stands for the front view, left view, back view, and right view, respectively.

gradient descent. We implement our method in PyTorch. We use Adam [36] with L2 regularization (weight decay $= 10^{-7}$) and set the learning rate $= 10^{-4}$. We re-scale the pixel value of the RGB observations from $[0, 255]$ to $[0, 1]$ and use binary cross-entropy loss. We train the local occupancy predictor with 4 random seeds. For each seed, we randomly split the navigation dataset into training, validation, and test sets.

C. *Localizing the robot on the coarse map*

Using the predicted local occupancy as the observation, CMN needs to estimate the robot's location on the coarse map. Importantly, we directly treat the original coarse map as a discrete occupancy grid. Defined in Section IV-A, the coarse map is a different representation of the real world with an unknown diffeomorphism operator $\mathcal{F}_d$. However, CMN does not estimate the diffeomorphism operator due to the lack of the robot's positional information. Instead, CMN makes a simple modification to a discrete Bayesian filter [37] to localize the robot on the coarse map. In particular, we argue that the predicted local occupancy $\hat{m}$ contains rich information to correct errors during the belief update.

First, the coarse map image is converted into a binary occupancy grid where each pixel corresponds to one grid cell with value 1 if occupied, and 0 if empty. Given this binary occupancy grid, we assume that the robot's initial location is unknown but that the goal is marked on the grid. Therefore, CMN starts with a uniform belief $\widehat{Bel}(\hat{s}_0)$ over the entire state space $\hat{s} \in \hat{\mathcal{S}}$ on the grid/map. Suppose the belief is $\widehat{Bel}(\hat{s}_t)$ at time step t, the robot takes action a_t and observes o_{t+1} in the real world. We explain the modified Bayesian filtering on the coarse map as follows.

Computing the predictive belief with a "noisy" transition update: We assume the robot's motion is deterministic in the real world. Specifically, the transition probability $p(s'|s, a)$ is 1, if s' is the resulting state after the robot takes a at state s, and 0 otherwise. Therefore, if the map is strictly aligned with the real world (i.e., $\mathcal{S} = \hat{\mathcal{S}}$), updating the predictive belief on the map is easy. However, when the map is coarse ($|\hat{\mathcal{S}}| \ll |\mathcal{S}|$), there exists an unknown relationship between the predictive belief update on the coarse map and the motion of the robot in the real world. Intuitively, one state $\hat{s}$ on the coarse map corresponds to a region in the real world of unknown size. In other words, the robot might remain in the same state on the coarse map even though it moves across several states in the real world. To tackle this issue, CMN proposes a "noisy" transition update. Specifically, when the robot takes an action a in the real world, we assume the robot will move to the next deterministic state s' on the map with probability p and remain in the same state s with probability $1-p$. Therefore, CMN performs the following "soft" update:

$$\widehat{Bel}^{-}(\hat{s}_{t+1} = \mathsf{s}') = p \cdot \widehat{Bel}(\hat{s}_t = \mathsf{s}) + (1-p) \cdot \widehat{Bel}(\hat{s}_t = \mathsf{s}') \tag{1}$$

Instead of using a fixed p, CMN randomly samples p from a uniform distribution between $[0, 1]$ because p is unknown and varies in different coarse maps. The randomly sampled p can be considered as an inductive bias representing how likely the robot remains in the same grid cell on the coarse map after taking one forward action in the real world. Thus, we call the update in Equation 1 a "noisy" transition update.

Correcting the predictive belief with an estimated measurement model: Although the predictive belief $\widehat{Bel}^{-}(\hat{s}_{t+1})$ is noisy, it roughly estimates the robot's motion. Furthermore, the predicted local occupancy grid $\hat{m}$ contains rich information to correct the belief. Therefore, the noisy predictive belief $\widehat{Bel}^{-}(\hat{s}_{t+1})$ can be corrected using the measurement model. The measurement model on the map is defined as $\hat{p}(m_{t+1}|\hat{s}_{t+1}, \hat{a}_t)$, where m_{t+1} is the observation of $\hat{s}_{t+1}$ after taking $\hat{a}_t$. Specifically, $m \in \hat{\Omega}$ is an $n \times n$ local occupancy grid on the map. However, the robot observes a panoramic RGB observation o_{t+1} in the real world rather than the local occupancy grid m_{t+1} on the map. Thus, the local occupancy predictor from Section IV-B is used to predict the local occupancy grid $\hat{m}_{t+1} = \mathcal{F}_\psi(o_{t+1})$. Finally, we propose a measurement model proportional to a normalized similarity score between $\hat{m}_{t+1}$ and m_{t+1}:

$$\hat{p}(m_{t+1}|\hat{s}_{t+1}, \hat{a}_t) \propto \left(1 - \frac{|\mathcal{F}_\psi(o_{t+1}) - m_{t+1}|}{n^2}\right) \tag{2}$$

Note that the scale mapping between coarse map and environment is not explicitly considered, though it is learned by the local occupancy predictor (implicitly via the training data); addressing this is future work. Using the estimated measurement model, we can correct the predictive belief:

$$\widehat{Bel}(\hat{s}_{t+1}) \propto \hat{p}(m_{t+1}|\hat{s}_{t+1}, \hat{a}_t)\, \widehat{Bel}^{-}(\hat{s}_{t+1}) \tag{3}$$

D. *Planning a local action using the global heuristic*

Given the latest belief, we first use it to extract a global heuristic that indicates a rough direction toward the goal. The state $\hat{s}$ with the highest probability is selected to be the location estimate (ties are broken randomly). Next, the A^* search algorithm [38] is applied to find the shortest path $\mathcal{L} = \{\hat{s}, \hat{s}_1, ..., \hat{s}_g\}$ between the location estimate $\hat{s}$ and the goal $\hat{s}_g$. Using the first two states $\hat{s}, \hat{s}_1 \in \mathcal{L}$, the heuristic is computed as $h = \hat{s}_1 - \hat{s}$ and is normalized as a unit vector.

Given the heuristic vector h, we propose a tree-structured planner. The key idea is to select the action that moves the agent in a similar direction indicated by h in the real world. Specifically, we build a k-step lookahead tree where the k-th layer is expanded by applying each action in the action space for each node in the $(k-1)$-th layer. Consider a planner with k steps, it results in a search tree with $|\mathcal{A}|^k$ leaf nodes. For each leaf node, we can compute a normalized directional vector. Next, we compute the cosine similarity between each leaf directional vector and the heuristic vector. We select the leaf node with the maximal cosine similarity and then traverse to the root node. The immediate action after the root node is selected as the best action to execute. Since all the leaf directional vectors can be computed relatively, CMN does not require any absolute robot's location as input.

V. EXPERIMENTS

To systematically demonstrate the effectiveness of CMN, we evaluate it on a simulated indoor visual navigation task using both *coarse-grid* and *hand-drawn* maps. In each navigation trial, the robot is initialized at a random unknown location in an environment it has not encountered before. The robot is given a map of the environment, either coarse-grid or hand-drawn, with the goal marked on it. The robot is tasked with navigating to the goal location using egocentric visual observations, its orientation, and the provided coarse map. We define three tasks with increasing challenges by changing the provided map types and sensor inputs.

Table I shows a summary of the evaluated navigation tasks. We evaluate most extensively on **Task CI** (Coarse-grid map with Image observations), where a coarse-grid map is automatically generated from the simulator, and egocentric visual observations are provided to the robot. We also evaluate on a smaller set of hand-drawn maps in **Task HD** (Hand-Drawn maps). Finally, to isolate the effect of perception, we consider **Task CG** (Coarse-grid map with Grid observations), where we directly provide the true local occupancy from the coarse map surrounding the robot, instead of visual observations. In this case, we skip the perception step from Section IV-B and directly use the provided local occupancy instead.

TABLE I

TASK VARIANTS AND CHALLENGES INVOLVED

	Task CG	Task CI	Task HD
Coarse-grid map	√	√	
Hand-drawn map			√
Local grid observations	√		
Image observations		√	√
Location uncertainty	√	√	√
Motion uncertainty	√	√	√
Observation uncertainty		√	√
Non-uniform scale (MPP)			√

A. Simluation environment

Habitat [9] is a photo-realistic simulator for embodied agent navigation. In Habitat, each environment is reconstructed using the RGB images captured from real houses. Habitat can seamlessly integrate different datasets to generate visually rich 3-D environments for indoor navigation. In this work, we use the environments from the Gibson [39] dataset, which contains 45 houses with different sizes and interior appearances. We randomly sample 33 houses to construct the offline dataset to train the deep perception model and hold out 9 unseen environments for navigation evaluation. Note that the collected offline dataset is exclusively used for training the local-occupancy predictor (Section IV-B), with no navigation-related training involved.

Obtaining coarse maps: Each Habitat environment has a pre-built metric occupancy map. To obtain *coarse-grid* maps, we downsample the metric occupancy maps using Habitat's built-in functions. The coarseness of the maps is controlled by the *Meters Per Pixel* (MPP) parameter. In this work, we examine MPP = 0.3, 0.4, and 0.5. MPP > 0.5 causes excessive distortion and are not usable. For example, MPP $= 0.3$ means one pixel on the map represents a 0.3×0.3 m^2 region in the real environment. To generate *hand-drawn* coarse maps, a human operator observes the top-down 3-D views of the environments and draws the maps.

Local observations: We consider two types of local observations: (1) **Grid** observation: 3×3 occupancy grid cropped from the map, and (2) **Image** observation: $4 \times 80 \times 80 \times 3$ RGB images captured from the four directions (front, back, left, right), forming a panoramic view.

Robot actions: The robot has three available actions: {move_forward, turn_left, turn_right}. move_forward advances the robot by 0.15 m. The turning actions will turn the robot 90° in the respective directions.

Navigation episode: Each navigation episode has a maximum horizon of 500 time steps. An episode will terminate automatically if the distance between the robot and the goal location is smaller than a threshold value (success) or it reaches the maximal time horizon (failure).

B. Evaluation metrics

To measure CMN's performance, we use the mean Success Rate (SR) and the mean Success rate weighted by Path Length (SPL) [40] as evaluation metrics. We sample $N = 50$ episodes with arbitrary start and goal locations in each test environment. The mean *SR* is defined as $\frac{\sum S_i}{N}$, where S_i equals to 1 if the i-th episode is successful and 0 otherwise. The mean *SPL* is defined as $\frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max(l_i, p_i)}$, where l_i is the length of the shortest path distance and p_i is the length of the actual path taken by the agent in the i-th episode. We report the final results by averaging the mean *SR* and mean *SPL* over all test environments.

C. Compared approaches

We compare our approach, Coarse Map Navigator (**CMN**), against two baselines. The first method, Monte-Carlo Scale Estimation (**MCSE**), is based on the approach in [5], [6], where a particle filter is used to estimate the scale of local deformations in the coarse map. This estimate informs the transition probabilities and the observation model. The second method, Rescaled Map Navigator (**RMN**), scales the coarse map to the size of the true environment, assuming that this scale is provided. The resulting rescaled map is similar in dimension to a classical metric map, but the features may be less smooth and rougher due to the coarsening operator. The scale parameter (i.e., MPP) is not typically available, but we provide this privileged information to **RMN** only. We use

TABLE II

Task CI: Coarse-grid maps with image observations

	MPP $= 0.3$		MPP $= 0.4$		MPP $= 0.5$	
Method	SR	SPL	SR	SPL	SR	SPL
RND	10.7%	9.5%	10.7%	9.5%	10.7%	9.5%
RMN	53.3%	45.8%	54.1%	11.4%	49.3%	11.3%
MCSE	78.9%	28.4%	76.9%	28.8%	**70.0%**	30.4%
CMN	**91.6%**	**50.7%**	**87.8%**	**51.2%**	**70.0%**	**37.7%**

TABLE III

Task CG: Coarse-grid maps with grid observations

	MPP $= 0.3$		MPP $= 0.4$		MPP $= 0.5$	
Method	SR	SPL	SR	SPL	SR	SPL
RMN	70.4%	33.4%	63.0%	29.1%	57.8%	26.2%
CMN	**94.7%**	**58.0%**	**95.7%**	**57.3%**	**83.8%**	**51.9%**

RMN to illustrate how a standard navigation pipeline using a metric map may behave when using a coarse map instead.

To our knowledge, there is no visual navigation method that takes in visual observations and outputs a coarse local occupancy grid, so all the compared methods use our proposed local occupancy predictor (except in **Task CG**). Additionally, for fairness and ease of comparison, all compared approaches use the same planner as CMN.

Finally, we also include a Random policy (**RND**) to show that the evaluated navigation tasks are non-trivial.

D. Results

Table II shows that **CMN** outperforms **RMN** by a significant margin across various MPP scales, demonstrating the effectiveness of our proposed framework. We hypothesize that **CMN** variants are better because of two reasons. (1) No scaling: The coarse maps already contain some inaccuracies, so scaling them up for **RMN** might further amplify these inaccuracies. (2) Smaller belief space: Although coarse maps are rough, they shrink the size of the belief space over potential robot locations.

Somewhat surprisingly, **MCSE** also performs worse than **CMN**, even though **MCSE** attempts to estimate the local scale parameter, whereas **CMN** essentially samples the scale at random (in the "noisy" transition update). We hypothesize that the local scale may actually be quite difficult to estimate accurately, especially when using visual observations. **MCSE** was only previously evaluated on laser rangefinder data, which may be more appropriate for scale estimation.

Overall, **CMN** is also significantly better in SPL, indicating that not only does it achieve greater navigation success, it does so more efficiently compared to other methods.

Table III shows the performance when image observations are replaced with the true local coarse occupancy grid, thereby removing the perception (local occupancy prediction) component. The task becomes significantly easier and **CMN** is able to achieve high success rates. Although **RMN** performance also improves significantly, the gap is still large compared to **CMN**, suggesting that simply rescaling the coarse map is not an adequate approach, even if the scale (MPP) was known and perceptual ambiguity was removed.

Finally, we consider a more realistic setup in Task HD, where coarse maps are drawn by non-expert human beings. A human operator draws the coarse map by observing a 3-D top-down view of an environment. The *hand-drawn* maps are different from the *coarse-grid* maps because they introduce scale inconsistency. For **RMN**, we still resize the *hand-drawn* occupancy maps to the ground-truth size; for **CMN**, we directly use the provided *hand-drawn* maps (e.g., as a scanned image). CMN still achieves 75.6% navigation success rate using *hand-drawn* maps (see Table IV).

TABLE IV

Task HD: Hand-drawn maps with image observations

Method	SR	SPL
RMN	58.5%	23.8%
CMN	**75.6%**	**36.5%**

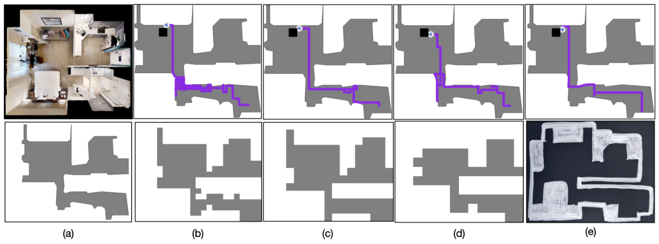

Fig. 5. Visualization of trajectories for CMN (top row) using different coarse maps (bottom row). Column (a) shows the top-down view and the ground-truth *metric* map of one environment. Columns $(b)-(d)$ show the trajectories of the robot using *coarse-grid* maps with MPP $= 0.3, 0.4, 0.5$ respectively (input coarse maps shown in bottom row). Column (e) shows the trajectory of the robot using a *hand-drawn* map.

Figure 5 shows an example of CMN navigating using various types of coarse maps. Please see the accompanying video for more examples.

VI. Conclusion

In this work, we propose CMN, a fully autonomous navigation system that can use different coarse maps and visual input to perform robot navigation in previously unseen environments. In CMN, the proposed local occupancy predictor makes reliable predictions from visual inputs and facilitates quick generalization to new environments. Using predicted local occupancy grids as observations, the modified Bayesian filter enables CMN to tackle errors and misalignment between the coarse map and the real world. We demonstrate the performance of CMN through systematic experiments, which provide empirical insight into CMN.

In the future, we will consider several extensions to CMN. For perception, we will reduce the number of required cameras and use just the forward-facing camera as visual input, which requires improvements to the local occupancy predictor. We will also consider more challenging coarse and incomplete maps. For instance, coarse maps that only contain walls, but not furniture, will require a more sophisticated belief model to tackle the increasing localization difficulty caused by the missing furniture information. Finally, since indoor environments are usually dynamic and populated by humans, we envision extending CMN to social navigation.

Acknowledgment

We thank Kevin Robb for extending and validating CMN on a physical robot in his M.S. thesis [41], [42].

References

[1] S. G. Tzafestas, “Mobile robot control and navigation: A global overview,” *Journal of Intelligent & Robotic Systems*, vol. 91, no. 1, pp. 35–58, 2018.

[2] X. Xiao, B. Liu, G. Warnell, and P. Stone, “Motion planning and control for mobile robot navigation using machine learning: a survey,” *Autonomous Robots*, pp. 1–29, 2022.

[3] H. P. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *IEEE International Conference on Robotics and Automation*, 1985.

[4] D. Filliat and J.-A. Meyer, “Map-based navigation in mobile robots:: I. a review of localization strategies,” *Cognitive Systems Research*, vol. 4, no. 4, pp. 243–282, 2003.

[5] B. Behzadian, P. Agarwal, W. Burgard, and G. D. Tipaldi, “Monte Carlo localization in hand-drawn maps,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[6] F. Boniardi, B. Behzadian, W. Burgard, and G. D. Tipaldi, “Robot navigation in hand-drawn sketched maps,” in *European Conference on Mobile Robots*, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, “Bayesian filtering for location estimation,” *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24–33, 2003.

[9] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, “Habitat: A platform for embodied AI research,” in *IEEE/CVF International Conference on Computer Vision*, 2019.

[10] G. N. DeSouza and A. C. Kak, “Vision for mobile robot navigation: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 237–267, 2002.

[11] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.

[12] P. Karkus, D. Hsu, and W. S. Lee, “Particle filter networks with application to visual localization,” in *Conference on Robot learning*, 2018.

[13] S. K. Gottipati, K. Seo, D. Bhatt, V. Mai, K. Murthy, and L. Paull, “Deep active localization,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4394–4401, 2019.

[14] D. S. Chaplot, E. Parisotto, and R. Salakhutdinov, “Active neural localization,” *International Conference on Learning Representations*, 2018.

[15] X. Ma, P. Karkus, D. Hsu, and W. S. Lee, “Particle filter recurrent neural networks,” in *AAAI Conference on Artificial Intelligence*, 2020.

[16] G. Brunner, O. Richter, Y. Wang, and R. Wattenhofer, “Teaching a machine to read maps with deep reinforcement learning,” in *AAAI Conference on Artificial Intelligence*, 2018.

[17] P. Karkus, X. Ma, D. Hsu, L. P. Kaelbling, W. S. Lee, and T. Lozano-Pérez, “Differentiable algorithm networks for composable robot learning,” *Robotics: Science and Systems*, 2019.

[18] S. Koenig and R. G. Simmons, “Passive distance learning for robot navigation,” in *International Conference on Machine Learning*, 1996.

[19] K. Konolige, E. Marder-Eppstein, and B. Marthi, “Navigation in hybrid metric-topological maps,” in *IEEE International Conference on Robotics and Automation*, 2011.

[20] V. Setalaphruk, A. Ueno, I. Kume, Y. Kono, and M. Kidode, “Robot navigation in corridor environments using a sketch floor map,” in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2003.

[21] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, “A tutorial on graph-based SLAM,” *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[22] M. Skubic, S. Blisard, A. Carle, and P. Matsakis, “Hand-drawn maps for robot navigation,” in *AAAI Spring Symposium on Sketch Understanding*, 2002.

[23] G. Chronis and M. Skubic, “Sketch-based navigation for mobile robots,” in *IEEE International Conference on Fuzzy Systems*, 2003.

[24] M. Skubic, C. Bailey, and G. Chronis, “A sketch interface for mobile robots,” in *IEEE International Conference on Systems, Man and Cybernetics*, 2003.

[25] M. Skubic, S. Blisard, C. Bailey, J. A. Adams, and P. Matsakis, “Qualitative analysis of sketched route maps: translating a sketch into linguistic descriptions,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 2, pp. 1275–1282, 2004.

[26] F. Boniardi, A. Valada, W. Burgard, and G. D. Tipaldi, “Autonomous indoor robot navigation using a sketch interface for drawing maps and routes,” in *IEEE International Conference on Robotics and Automation*, 2016.

[27] J. Yun and J. Miura, “A quantitative measure for the navigability of a mobile robot using rough maps,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[28] M. Mielle, M. Magnusson, and A. J. Lilienthal, “Using sketch-maps for robot navigation: Interpretation and matching,” in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2016.

[29] G. Parekh, M. Skubic, O. Sjahputera, and J. M. Keller, “Scene matching between a map and a hand drawn sketch using spatial relations,” in *IEEE International Conference on Robotics and Automation*, 2007.

[30] K. Matsuo and J. Miura, “Outdoor visual localization with a hand-drawn line drawing map using FastSLAM with PSO-based mapping,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[31] X. Wang, R. J. Marcotte, and E. Olson, “GLFP: Global localization from a floor plan,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[32] Z. Li, M. H. Ang, and D. Rus, “Online localization with imprecise floor space maps using stochastic gradient descent,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[33] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *National Conference on Artificial Intelligence*, 2002.

[34] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Planning and acting in partially observable stochastic domains,” *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[35] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, “Learning to explore using active neural SLAM,” *International Conference on Learning Representations*, 2020.

[36] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations*, 2015.

[37] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: The MIT Press, 2005.

[38] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[39] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, “Gibson Env: Real-world perception for embodied agents,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[40] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, and A. R. Zamir, “On evaluation of embodied navigation agents,” *arXiv preprint arXiv:1807.06757*, 2018.

[41] K. M. Robb, “Mobile robot localization and navigation in physical environments using hand-drawn maps,” Master’s thesis, Northeastern University, 2023.

[42] K. Robb, “Implementation of coarse map navigation (CMN) on a physical robot,” https://github.com/kevin-robb/coarse-map-nav-integration.