Indoor Navigation for Mobile Agents: A Multimodal Vision Fusion Model

Dongfang Liu[1]*, Yiming Cui[2]*, Zhiwen Cao[1], and Yingjie Chen[1]
[1]Department of Computer Graphics Technology, Purdue University, West Lafayette, 47907, USA
Email: {liu2538, cao270, victorchen}@purdue.edu
[2]Department of Electrical & Computer Engineering, University of Florida, Gainesville, FL, 32611, USA
Email: cuiyiming@ufl.edu
* authors contribute equally on this work.

***Abstract*—Indoor navigation is a challenging task for mobile agents. The latest vision-based indoor navigation methods make remarkable progress in this field but do not fully leverage visual information for policy learning and struggle to perform well in unseen scenes. To address the existing limitations, we present a multimodal vision fusion model (MVFM). We implement a joint modality of different image recognition networks for navigation policy learning. The proposed model incorporates object detection for target searching, depth estimation for distance prediction, and semantic segmentation to depict the walkable region. In design, our model provides holistic vision knowledge for navigation. Evaluation on AI2-THOR indicates that MVFM improves on the results of a strong baseline model by 3.49% for Success weighted by Path Length (SPL) and 4% for success rate respectively. In comparison with other state-of-the-art systems, MVFM performs in the lead in terms of SPL and success rate. Extensive experiments show the effectiveness of the proposed model.**

***Index Terms*—Visual navigation, object detection, depth estimation, semantic segmentation**

I. Introduction

Indoor navigation is an essential capability of mobile robotic systems. To complete different tasks, it is important for a mobile robot to effectively search, locate, and reach an arbitrary object in an indoor environment [1] [2]. Following human order, the mobile robot should be able to navigate toward designated objects or regions in an indoor environment efficiently. There are divergent approaches to indoor navigation. A large array of work for navigation focuses on motion planning which requires a barrier-free path in the configuration space or workspace, with clear geometric information of the testing environment [3]. However, path planning and low-level control for this motion planning approach usually assume perfect localization based on a high-quality geometric construction of the environment. This limits the generalization of these methods [4]. Besides, a rich and informative body of work has explored Simultaneous Localization and Mapping (SLAM) [6] for indoor navigation. Mobile robots can use SLAM to have a global perception of the environment and localize areas and objects. SLAM, however, needs to employ geometric techniques and construct metric maps, while navigation for a mobile robot is not the primary consideration. Environmental representations built by SLAM systems are often not compatible with reliable indoor navigations because the environmental settings may often alter over time in practice.

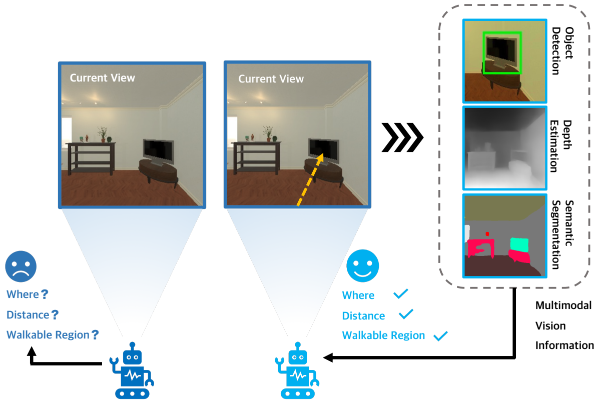


Fig. 1: A holistic vision knowledge for navigation. On the left, an active agent needs know where the target is, how far the target is, and what the walkable region for navigation is. On the right, the multiple vision modalities are used for navigation which incorporates object detection for target searching, semantic segmentation for finding walkable regions, and depth estimation for distance prediction. Employing multimodal vision knowledge, the proposed model can accurately navigate an agent in an indoor environment.

In contrast, vision-based systems appear to construct more flexible representations because mobile robots can utilize imagery input to robustly navigate in an indoor environment without knowing the precise localization or using a metric map [5]. Visual navigation systems can leverage prior knowledge for path planning from previously seen environments [1] [2] [6]. With the recent development of deep learning approaches for computer vision, we have witnessed significant breakthroughs in reinforcement learning algorithms and convolutional neural networks which facilitate the improvement of robotic motor control [7] and visual perception systems [8] [9] [10]. Deep-learning-based vision models bridge the communication barrier between robotic perception and control with the convergence of solutions in both tasks. Recently, [4] [5] employed joint modeling of visual recognition and navigation policy learning to control a mobile robot to find a specific visual target in an indoor scene. However, the latest vision-based indoor navigation methods fail to fully uses vision information for policy learning (mainly object detection or segmentation). Thus, these methods learn limited visual knowledge for action or decision making and struggle to generalize for an unseen scene.

To address the limitations of previous work, we present a multimodal vision fusion model (MVFM) in this paper. We attempt to move a step further to develop a joint modality of multiple vision recognition networks and navigation policy learning to approach the task of creating a "robot with a vision that finds the object". Our multimodal vision fusion model includes object detection (to locate the target), depth estimation (to determine the distance), and semantic segmentation (to depict the walkable region), which collectively provide a holistic vision knowledge for navigation (shown in Fig. 1). With the visual information from multiple modalities, our model can accurately search objects with an arbitrary pose in any given location.

To study the aforementioned task, we leverage a simulation dataset as a benchmark for training and testing. The simulation dataset is collected from the AI2-THOR challenge platform [11], which provides a high-realism environment as a testbed. Contributions of our work include :

- The proposed model can control a mobile agent in navigating to a given target in an indoor environment using only visual observations.
- The proposed model effectively integrates multiple image recognition modalities in vision-guided action policy learning, offering a more holistic visual knowledge for navigation. We have not found any previous work using this approach. With the assistance of multimodal vision fusion, the proposed model improves on the results of a strong baseline model by 4.68% for SPL and 5.37% for success rate respectively.
- We conduct extensive experiments to compare our model with the latest state-of-the-art systems. Results indicate the proposed model is competitive with the best existing approaches.
- We introduce a simulation dataset from the AI2-THOR platform to benchmark for future research in this task.

II. Related Work

A. Deep Models for Indoor Navigation

Path planning for traditional indoor navigation methods typically bears an assumption that the environmental map is given or constructed while the exploration proceeds [12]. In contrast, [13] [14] [15] recently explored learning-based navigation approaches for performing localization, mapping, and exploration end-to-end. These approaches have achieved recognizable results and outperformed traditional indoor navigation. For instance, [15] designed an indoor navigation system by giving the mobile robot a picture of the target to search; [14] addressed the challenge of indoor navigation by training a joint mapper and planner model; and [13] used loop closure to speed up RL training for indoor navigation. [16] used topological maps for the indoor navigation task. To construct the maps, they prolonged the exploration of the test environment to populate the navigation memory. Different from the above work, our model can navigate without a map; it relies solely on vision clues. [17] designed a self-supervised deep RL model for navigation. However, no semantic information is used for this model. To improve navigation performance, [18] employed object detectors and semantic segmentation modalities to predict navigation policies. Similarly, [19] [20] incorporated semantic knowledge to improve the generalization of unseen scenarios. These previous studies impact our work.

B. Vision-Based Indoor Navigation

Vision-based perception for indoor navigation has gained popularity in recent years. [4] [15] proposed an effective approach for a mobile robot to search for a target object based on solely visual inputs in indoor environments. They employed reinforcement learning to learn the relationship between the camera input of the current state and the actions of policy to reach the vicinity of the searched object or to match a specific scene. For better generalization, [15] [4] utilized scene-specific layers in their model to robustly navigate in new scenes. In the same vein, [21] proposed several incremental extensions for the scene recognition modality in their approach, but the result indicated a limited contribution toward improving visual navigation performance. For a target-driven search, [4] [15] [21] designated a specific scene image for a robot to use to navigate to the place where the target image was taken. This approach limits the potential for practical application because the target image needs to be available for inference. In the present work, our proposed model can search and locate any object without having access to a target image.

Wortsman et al [5] proposed a meta-learning approach to integrating object recognition and a Long Short-Term Memory network (LSTM) into a unified framework. Results from [5] showed that with the LSTM-supported algorithm, the proposed model can achieve better policy learning and a model using meta-learning can more effectively navigate new scenes. Similarly, [22] employed meta-learning to encourage mobile robots to explore the state space outside the dictated actors' policy, which significantly improved training efficiency and robustness. [23] utilized meta-learning to augment the agent's policy by adding structured noise in training, so the agent can more robustly navigate in unseen episodes by inference. However, for meta-learning based indoor navigation, the proposed model needs to constantly update a large number of parameters during navigation, which requires extra memory and causes computation overloads. Compared to [22] [23], our proposed model has fewer parameters to update during inference. Hence, our proposed model achieves a faster runtime performance.

Inspired by previous work, we propose a novel vision-based model that utilizes visual knowledge from multiple modalities and constructs a holistic vision view for training and inference. We leverage reinforcement learning to learn the relationship between the current view and the action policy to navigate a mobile agent. With the newly-designed vision recognition network, our model can accurately search out objects with an arbitrary pose in any given location and robustly navigate a mobile robot within an unseen indoor environment.

III. Multimodal Vision Fusion Model

A. Task Definition and Overview

Our proposed model is trained to learn action policies for the active agent to navigate to the target using only image sequences from a monocular camera. Following the design from [5], we denote $\mathcal{I} = \{I_1, \ldots, I_t\}$ as a sequence of camera images, $\mathcal{O} = \{o_1, \ldots, o_m\}$ as object classes for different targets, and p as the position of the agent. For each task $\tau \in \mathcal{T}$, we denote each task $\tau = (\mathcal{I}, o, p)$. We separate different scenes for training $\mathcal{T}_{train}$ and testing $\mathcal{T}_{test}$ tasks. Each trial of navigation for a task is an episode in our work.

In our work, we use Glove embedding to specify the target object class [24] and the agent is required to use only the monocular RGB inputs to navigate to the target object. Navigation is built on a sequence of actions which are $\mathcal{A} = \{$MoveAhead, RotateLeft, RotateRight, LookDown, LookUp, MoveBack, Done$\}$. For each step, the horizontal rotation has maximum 45 degrees while the first-person view can incline maximum 30 degrees. Before reaching the target, the agent takes an action from action set $\mathcal{A}$. If the target object is found, the agent will take a termination action. An episode is counted as successful if the agent uses the designated steps to reach the given target in a close range (e.g. within 1 meter of the agent) and issue a termination action. Otherwise, the episode concludes as a failure case and the navigation task is unsuccessful.

B. Model Design

Modern CNN-based image recognition networks share a similar structure [25] [26] [27] [28]. In general, a backbone sub-network is employed on the input image to produce feature maps on the whole image. Then, a shallow sub-network for a specific task is applied on the feature maps to generate the output. Bearing this in mind, we propose a novel deep reinforcement learning-based framework which integrates four basic modalities.

The framework of the proposed model is demonstrated in Fig. 2. Taking RGB images from the camera, we employ a feature extraction modality $\mathcal{M}_{ext}$ as a backbone sub-network to produce feature maps f_{im} of the input images. The extracted feature maps are then fed into three different modalities for different dimensions of features: 1. a semantic segmentation modality $\mathcal{M}_{seg}$ for scene segmentation feature f_{seg}, 2. a depth estimation modality $\mathcal{M}_{dep}$ for depth feature f_{dep}; and 3. an object detection modality $\mathcal{M}_{det}$ for object detection feature f_{det}. Concatenating each piece of embedded information from each modality output, we obtain the joint feature map f_{joint} and perform a pointwise convolution. The output is then given as input to the Long Short-Term Memory (LSTM) modality $\mathcal{M}_{LSTM}$. The linear layer in $\mathcal{M}_{LSTM}$ produces the action policy and value of the active agent for navigation. The working pipeline for MVFM is summarized in Algorithm 1.

Algorithm 1 The Working Pipeline for MVFM

```
input: frame$\{I_t\}$                              ◁ The current view
while not a termination action do                  ◁ Not reach the target
    $f_{im} = \mathcal{M}_{ext}(I_t)$                      ◁ Extract image features
    $f_{seg} = \mathcal{M}_{seg}(f_{im})$                  ◁ Get segmentation features
    $f_{dep} = \mathcal{M}_{dep}(f_{im})$                  ◁ Capture depth features
    $f_{det} = \mathcal{M}_{det}(f_{im})$                  ◁ Extract detection features
    $f_{joint} = Concat(f_{seg}, f_{dep}, f_{det})$        ◁ Obtain joint features
    $action = \mathcal{M}_{LSTM}(f_{joint})$               ◁ Produce action policy
return: $action$
```

C. Network Architecture

For different recognition tasks, we adopt state-of-the-art architectures and craft them into our proposed model.

1) Feature Extraction Modality: We use ResNet-50 [29] which is pre-trained for ImageNet classification. The entirety of the ImageNet dataset is processed by image-to-image translation [30] with the images collected from the AI2-THOR platform so the pre-trained ResNet-50 is sensitive to AI2-THOR contexts. In order to align with the design from semantic segmentation [26], depth estimation [31], and object detection [25], we reduce feature stride from 32 to 16 to produce denser feature maps. We randomly initialize a 3×3 convolution where the holing algorithm [32] is employed to keep the field of view and append it to the first block of the conv5 layers to reduce the feature channel to 1024. Finally, we discard the last 1000-way classification layer, so the output 1024-d feature maps are fed into the subsequent modalities.

2) Semantic Segmentation Modality: We employ DeepLab [26] which is pre-trained on the ADE20K dataset [33] with style transferred to AI2-THOR context. We remove the original backbone network and randomly initialize the 1×1 convolutional layer, where the intermediate feature is processed to produce (C+1) score maps (C indicates the number of object categories and 1 is the background category). We remove the softmax layer which produces the per-pixel probabilities and replace it with a pointwise convolution layer to generate the scene segment embedding features. In our work, the scene segmentation modality only has two learnable weight layers.

3) Depth Estimation Modality: We employ monodepth [31] for our depth estimation modality. We remove the original encoder (from cnv1 to cnv7b) and use the feature map from the feature extraction network as input for the decoder (from upcnv7). We use the output before the disparity predictions and feed the depth features into a pointwise convolution layer to produce depth embedding features. The depth estimation modality is pre-trained by NYU's depth dataset [34] with style transferred to AI2-THOR context.

4) Object Detection Modality: We modify the state-of-the-art R-FCN [25] to obtain the object detection features. The RPN sub-network and the R-FCN sub-network are applied on the 1024-d feature maps. We adopt 9 anchors (3 scales

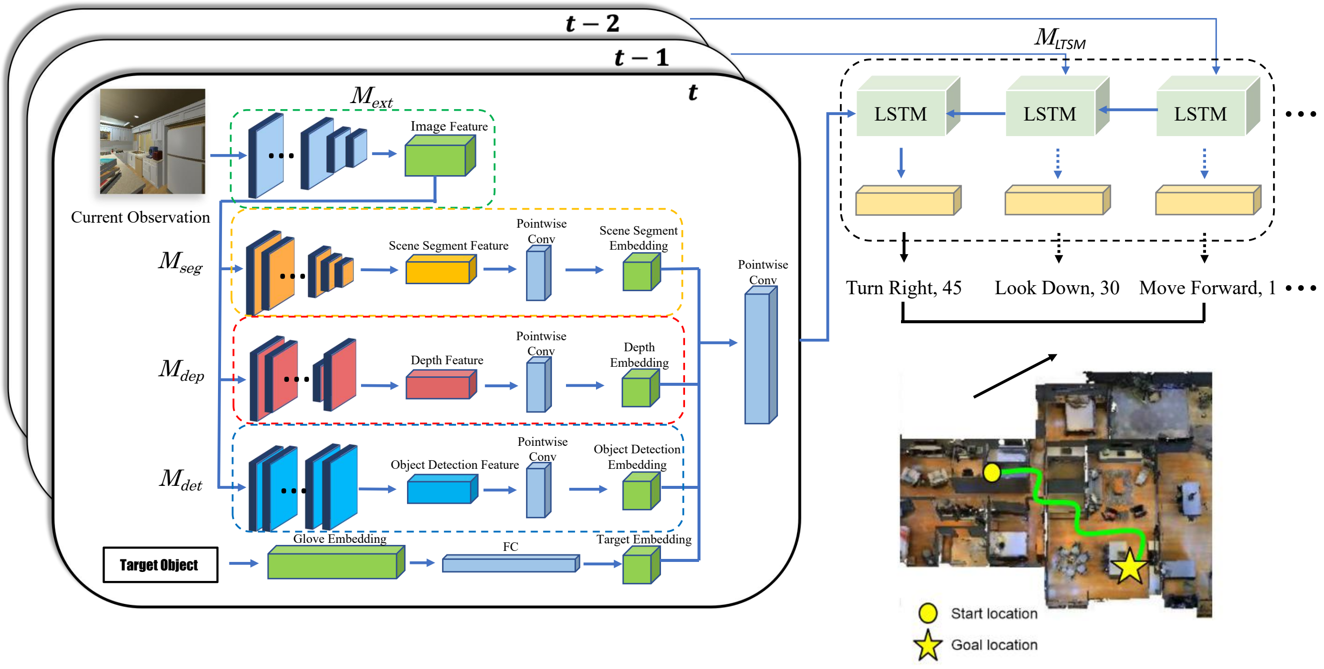


Fig. 2: The framework of the multimodal vision fusion model (MVFM). $\mathcal{M}_{ext}$ is the feature extraction modality; $\mathcal{M}_{seg}$ is the semantic segmentation modality; $\mathcal{M}_{dep}$ is the depth estimation modality; $\mathcal{M}_{det}$ is the object detection modality; and $\mathcal{M}_{LSTM}$ is the Long Short-Term Memory (LSTM) modality. Leveraging scene segmentation features, depth features, and object detection features, the proposed model considers multimodal vision knowledge in path planning, which improves the performance of the navigation.

and 3 aspect ratios) in RPN which can produce 300 proposals on each image, while the position-sensitive score maps in R-FCN are 7×7. We output before the objectness localization predictions and feed the detection features into a pointwise convolution layer to produce detection embedding features. The object detection modality is pre-trained with the COCO dataset [35] with style transferred to AI2-THOR context.

D. Learning Objective

At time t, the proposed method (parameterized by θ) processes the egocentric RGB image I_t and the target object class o, and outputs the distribution of the predicted actions which include $p_\theta(I_t)$ and a scalar $s_\theta(I_t)$. The distribution $p_\theta(I_t)$ is the predicted policy of the mobile agent while $s_\theta(I_t)$ is the value of the state. We use $p_\theta^a(I_t)$ to define the probability that the mobile agent would choose action a at time t.

Following the above overview of our proposed model, we discuss the learning objectives for the proposed model in this section. In vision-based navigation, much previous work has demonstrated that an active agent can learn and adapt to its environment by interacting with that environment [4] [15] [21]. Inspired by this previous work, we also want our model to handle obstacles by learning from prior knowledge based on environmental interaction. We therefore design a novel learning objective to facilitate the agent in learning to evolve from each interaction and in improving its navigation performance. Our learning objective stems from [36], which is given by:

$$\min_{\theta} \sum_{\tau \in \mathcal{T}_{train}} \mathcal{L}\left(\theta - \alpha \nabla_\theta \mathcal{L}\left(\theta, \mathcal{D}_\tau^{tr}\right), \mathcal{D}_\tau^{val}\right). \qquad (1)$$

where $\mathcal{L}$ is the overall loss which is defined by the network parameters θ and a function of a dataset. $\mathcal{T}_{train}$ is the entire set of tasks including the training $\mathcal{D}_{tr}$ and validation $\mathcal{D}_{val}$ datasets. α is the step-size learning rate and ∇ denotes the differential operator. The learning objective for Eq. (1) is to learn parameters θ so the trained model can adapt quickly even on novel tasks. Namely, an agent can optimize its performance on $\mathcal{D}_\tau^{val}$ after adapting to the task with a gradient step on $\mathcal{D}_\tau^{tr}$.

Following this design, our learning objective is that a mobile agent can continually learn from its interactions with the surrounding environment. In our design, Stochastic Gradient Descent (SGD) is used to update the model's adaptation of the environment, which modifies the policy network for the agent and facilitates its adaption to the scene. The SGD updates are associated with $\mathcal{L}_{int}$, which is the interaction loss in our work. We minimize $\mathcal{L}_{int}$ to optimize the mobile agent's actions to complete its navigation task. Thus, our learning objective is to learn θ for a good initialization, so the agent can leverage a few SGD updates using $\mathcal{L}_{int}$ and learn to adapt to the scene quickly. In inference, we employ a self-supervised loss for the agent to have access to $\mathcal{L}_{int}$. Our learning objective for the proposed model is writen in Eq. (2):

$$\min_{\theta} \sum_{\tau \in \mathcal{T}_{train}} \mathcal{L}_{nav}\left(\theta - \alpha \nabla_{\theta} \mathcal{L}_{int}\left(\theta, \mathcal{D}_{\tau}^{int}\right), \mathcal{D}_{\tau}^{nav}\right) \quad (2)$$

where $\mathcal{D}^{int}$ (for navigation task τ) denotes the internal state representations, observations, and actions for the travel trajectory of the agent, while $\mathcal{D}^{nav}$ denotes the remaining navigation trajectory. For very designated k steps of the mobile agent walking in the scene, an SGD update associated with the self-supervised loss is employed to obtain the adapted parameters $\theta - \alpha \nabla_{\theta} \mathcal{L}_{int}(\theta, \mathcal{D}_{\tau}^{int})$ to update θ. In this design, previously travelled scenes can help the agent adapt to new scenes with similar context.

In sum, we minimize our overall $\mathcal{L}_{nav}$ to maximize the reward for the active agent, to incentivize its actions in navigation to the target. The agent's policies, values, actions, and rewards throughout an episode rely on the function of the learning objective in our work.

IV. Experiment

We evaluate the proposed model on AI2-THOR platform. Extensive experiments demonstrate the efficacy of MVFM in finding the target object in an indoor context. All experiments are conducted on a workstation with 4 NVIDIA GTX 1080Ti GPUs and Intel Core i7-4790 CPU. Our experiments are elaborated below.

A. Experiment Platform

The AI2-THOR platform simulates photo-realistic indoor scenes for training mobile agent systems to search for and navigate to objects in virtual environments. Specifically, AI2-THOR includes four room categories, which are living room, kitchen, bathroom, and bedroom. Based on our design, the agent is designed to have 7 discrete actions, $\mathcal{A} = \{$MoveAhead, RotateLeft, RotateRight, LookDown, LookUp, MoveBack, Done$\}$. Based on the platform settings, a set of images is taken from each robot state for each indoor scene and the overall state space of the mobile agent includes the full set of images from each scene. A set of target images from each scene is also provided which is convenient for our experiments.

B. Implementation Details

In our experiment, we employ 22 scenes for training, 4 for validation, and 4 for testing for each room category, for a total of 120 scenes. For the navigation targets, a sequence of objects for each room category is selected based on their visibility and relative proportions within the entire image. Specifically, the selected targets are: fridge, toaster, microwave, bottle, coffee maker, box, trash can, and bowl in the kitchen; laptop, pillow, TV, box, vase, keychain, lamp, trash can, and bowl in the living room; ball, racket, laptop, cellphone, mug, plant, book, lamp, and alarm clock in the bedroom; and sink, soap bottle, toilet paper, towel, and toilet in the bathroom. For each scene, we randomly arrange the room settings and sample an object as a target with a random initial position.

In training, we arbitrarily stop the training when the success rate saturates on the validation set. We employ 10 asynchronous workers for training across all scenes. For $\mathcal{L}_{nav}$, we utilize -0.005 for taking each step and a reward of 4 for finding the object. In addition, we use SGD for interaction-gradient updates and Adam [37] for navigation-gradients.

In evaluation, we store parameters of each model during training and constantly test the performance of each stored model on our test dataset. We perform testing for 1200 different episodes, so each scene type has 300 episodes. For each scene in testing, we randomly initiate the initial state of the agent and the target object. All evaluations are conducted using the same testing sets.

C. Evaluation Method

We evaluate our method using both Success Rate and Success weighted by Path Length (SPL) [1].

1) Success Rate: We count completed tasks in the form of a binary indicator proposed by [1]. A successful task is counted as 1 while a failed task is 0. A navigation task is considered to be successful if the agent issues a termination action within one meter of the target object and the target is inside of the agent's camera view. However, if the agent's steps exceed 5000 steps without issue of a valid termination action, this episode is counted as a failure [4]. Let S_i be a binary indicator of success in episode i. The success rate is:

$$S_r = \frac{1}{N} \sum_{i=1}^{N} S_i. \quad (3)$$

2) SPL: [1] defines the SPL measure of the agent's navigation performance across the testing set as follows:

$$SPL = \frac{1}{N} \sum_{i=1}^{N} S_i \frac{l_i}{\max\left(p_i, l_i\right)}. \quad (4)$$

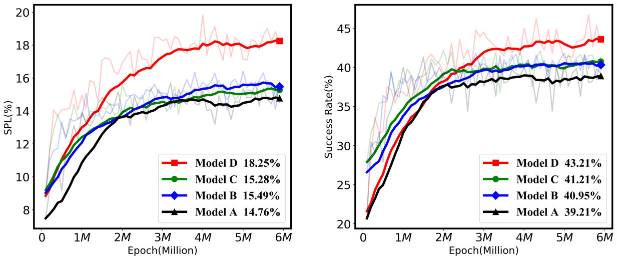
where S_i is the success indicator for the episode i; l_i is the shortest distance from the agent's initial position to the target position for the episode, and p_i is the actual length of the distance walked by the agent in the episode.

D. Experimental Results

1) Quantitative Results: For quantitative analysis, we conduct an ablation study to evaluate the improvements of the proposed model from baseline. We implement four models for comparisons to illuminate contributions from each added modality in the proposed model: Model A: The baseline model [4] which only includes an object detection feature for navigation policy learning; Model B: The baseline model [4] which includes object detection and segmentation features for navigation policy learning; Model C: The baseline model [4] which includes object detection and depth features for navigation policy learning; Model D: Our full model with object detection, segmentation, and depth features for navigation policy learning. We store the parameters of each model on every $10,000^th$ epoch of training and test the performance of the stored models on testing datasets. Fig. 3 depicts the pattern of testing results on each stored model through training.

Based on the results in Fig. 3, we observe an outstanding trend that adding depth and segmentation features to the baseline model (black line) can improve its performance on inference. For both SPL and success rates on the test dataset, the test results for model B (blue line) and model C (green line) are effectively increased after adding features from the other vision modality. Once training is saturated, SPL increases around 0.73% and 0.52% respectively for the final models B and C compared to model A; success rates of the final models B and C are also improved around 1.74% and 2.00% respectively compared with model A. However, the convergence speeds of models B and C, whether for SPL or success rate, decrease a little bit compared to model A.

Model D is the proposed model (red line), combining object detection features with depth and segmentation features. In Fig. 3(a), the SPL and success rate of the final model D are 18.25% and 43.10% respectively which is 3.49% and 4.00% higher than the baseline model and leads all the compared models. Also, the convergence speed of model D is faster than all the other models. Moreover, model D has a smaller standard deviation when the model converges compared with its counterparts, illustrated in Fig. 3(a) as background shadow lines. The results show the contributions of each added feature in the proposed model and the improvements in success rate offered by the proposed model. The proposed model achieves the best performance in all comparison to its counterparts.



(a) SPL (b) Success Rate

Fig. 3: Testing results through training. Model A is the baseline, models B and C are intermediate models, and model D is the proposed model. We store the parameters of a trained model on every $10,000^{th}$ epoch and test the saved model on testing datasets.

2) *Qualitative Analysis:* We present a qualitative analysis in this section. The directly visual comparison helps us to investigate the agent's performance, which sets our model apart from the baseline model. At the beginning of each episode, agents controlled by the baseline model and our model both intend to look around to locate the target object or explore the vicinity for visible clues (as shown in Fig. 4 and Fig. 5). While the episode proceeds, the baseline model might fail to circumvent obstacles or issue a wrongful termination action; with multimodal vision fusion knowledge, our model can constantly detect the target, determine the walkable region, and predict the distance of obstacles, so it seldom gets stuck behind an obstacle or fails to locate the target. Fig. 4 shows the trajectories of two episodes. Fig. 5 demonstrates egocentric views of each scene at selected steps. For the bedroom scene (where the target is an alarm clock), the baseline has trouble planning a feasible path to the target so it keeps running into the bed and wall for vision clues and eventually fails the mission. In contrast, our model marches to the target steadily because it locates the target at the beginning position, understands the walkable region, and knows how far the target is. Similarly, for the living room scene (where the target is a laptop), the baseline model successfully locates the target but it fails to circumvent the couch to reach the destination, while our model finds the target smoothly with its holistic vision knowledge. More examples can be found from https://youtu.be/uthSTrpZ04w.

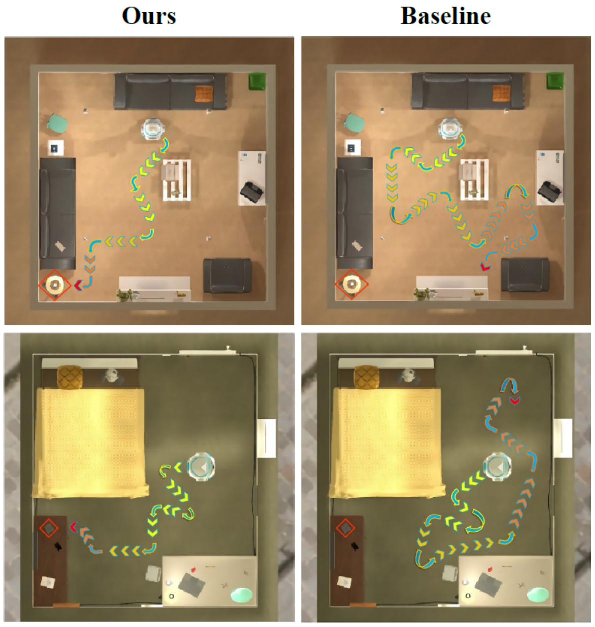


Fig. 4: Comparisons of trajectories. The target for the top scenes is lamp highlighted in orange and the target for the bottom scenes is alarm highlighted in red. The terminal action is marked as the red arrow.

E. Comparison with State-of-the-art Systems

We compare the performance of our model with the latest state-of-the-art models. In order to have a fair comparison, we select five leading models who also use AI2Thor as the testbed. The selected models and our model are all implemented on the same workstation and tested on our testing dataset for comparison.

Table I summarizes the results for all the models. Our model outperforms its counterparts in terms of SPL and success rate. We observe that most of the selected models do not fully

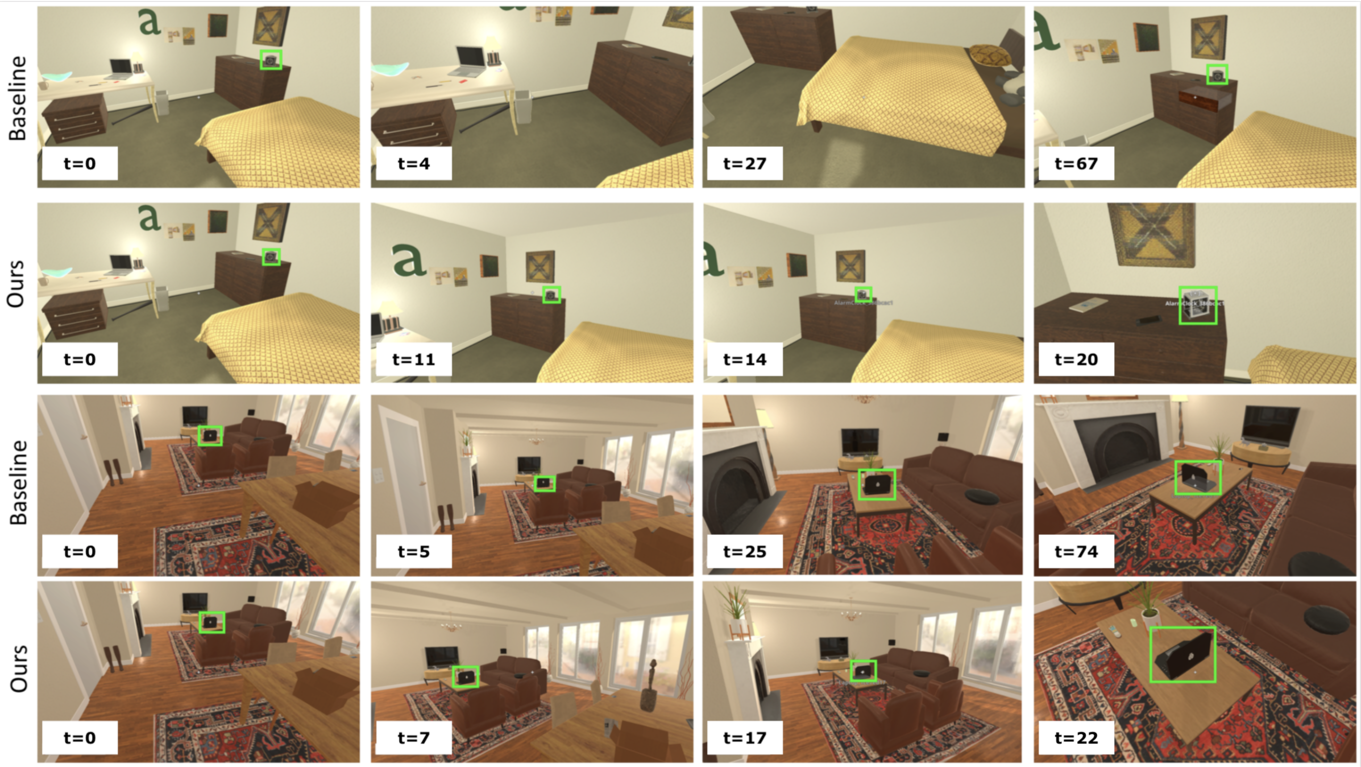


Fig. 5: Egocentric views of each scene at selected steps. The top two rows are a bedroom scene and the target is an alarm clock, while the bottom two rows are a living room scene and the target is a laptop. Our method outperforms the baseline for both tasks in terms of successful rate and SPL.

leverage the visual information to learn vision-guided action policy. In training and inference, TDRL [15], SP [20] and SAVN [5] only use a feature extraction network to explore vision clue, and RPLS [4] only uses object detection and feature extraction network to construct the image-action correlation. Their approaches are inferior to our model's, which employs multimodal (semantic segmentation, object detection, and depth estimation) vision fusion knowledge for vision-guided indoor navigation.

TABLE I: Comparison with State-of-the-Art Systems.

Method	SPL (%)	Success Rate (%)	Inference Time (s)
A3C [38]	11.68	30.04	0.31
TDRL [15]	12.47	31.01	0.32
RPLS [4]	14.76	39.21	0.43
SP [20]	15.47	35.39	0.37
SAVN [5]	16.15	40.86	0.81
Ours	**18.25**	**43.21**	**0.49**

We also report the inference time which is defined as the time it takes each model to generate an action at each step. A3C [38], a lightweight model for deep reinforcement learning, is leading all the models. Models using meta-learning approaches, such as SAVN [5] and our model, are slower than their counterparts. However, SPL and success rate for our model and SAVN [5] are much higher than A3C [38]. In addition, since our model has fewer parameters that need to be updated during inference compared to SAVN [5], our approach is significantly accelerated (as shown in Table I). Based on these results, we conclude that our model is highly competitive with the latest and best systems.

V. Conclusion

In this study, we presented a multimodal vision fusion model (MVFM) to fully employ visual information for policy learning. We developed a joint modality of different image recognition networks and navigation policy learning to approach the task of indoor navigation. Our multimodal vision fusion model included object detection (to locate the target), depth estimation (to determine the distance), and semantic segmentation (to depict the walkable region), which collectively provided a holistic vision knowledge for navigation. We conducted an ablation study to verify the effectiveness of each modality in our model. After comparing our model with other state-of-the-art systems, we argued that our model is competitive with the best existing approaches. Extensive studies indicated that the proposed model is effective for navigating a mobile agent through indoor environments in virtual reality. However, as the proposed model was observed only on the AI2Thor platform and with limited data, our results are more heuristic than general. This question remains open for further exploration in future works.

References

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

[2] D. Liu, Y. Wang, K. E. Ho, Z. Chu, and E. Matson, “Virtual world bridges the real challenge: Automated data generation for autonomous driving,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 159–164.

[3] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[4] X. Ye, Z. Lin, H. Li, S. Zheng, and Y. Yang, “Active object perceiver: Recognition-guided policy learning for object searching on mobile robots,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6857–6863.

[5] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, “Learning to learn how to learn: Self-adaptive visual navigation using meta-learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6750–6759.

[6] D. Liu, Y. Wang, and T. Chen, “Application of color filter adjustment and k-means clustering method in lane detection for self-driving cars,” in *2019 IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2019, pp. 153–158.

[7] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3d: Learning from rgb-d data in indoor environments,” *arXiv preprint arXiv:1709.06158*, 2017.

[8] R. A. Brooks and M. J. Mataric, “Real robots, real learning problems,” in *Robot learning*. Springer, 1993, pp. 193–213.

[9] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[10] Y. Wang, D. Liu, H. Jeon, Z. Chu, and E. T. Matson, “End-to-end learning approach for autonomous driving: A convolutional neural network model,” in *Proceedings of the 11th International Conference on Agents and Artificial Intelligence: ICAART*, 2019.

[11] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, “Ai2-thor: An interactive 3d environment for visual ai,” *arXiv preprint arXiv:1712.05474*, 2017.

[12] K. Kidono, J. Miura, and Y. Shirai, “Autonomous visual navigation of a mobile robot using a human-guided experience,” *Robotics and Autonomous Systems*, vol. 40, no. 2-3, pp. 121–130, 2002.

[13] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, “Cognitive mapping and planning for visual navigation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2616–2625.

[14] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu *et al.*, “Learning to navigate in complex environments,” *arXiv preprint arXiv:1611.03673*, 2016.

[15] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, “Target-driven visual navigation in indoor scenes using deep reinforcement learning,” in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[16] N. Savinov, A. Dosovitskiy, and V. Koltun, “Semi-parametric topological memory for navigation,” *arXiv preprint arXiv:1803.00653*, 2018.

[17] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, “Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[18] A. Toshev, A. Mousavian, J. Davidson, J. Kosecka, and M. Fiser, “Visual representations for semantic target driven navigation,” *ICRA*, 2019.

[19] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, and Y. Tian, “Learning and planning with a semantic model,” *arXiv preprint arXiv:1809.10842*, 2018.

[20] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, “Visual semantic navigation using scene priors,” *arXiv preprint arXiv:1810.06543*, 2018.

[21] L.-H. Chen, M. P. S. Moorthy, P. Sharma, and P. Kawthekar, “Imitating shortest paths for visual navigation with trajectory-aware deep reinforcement learning,” 2017.

[22] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine, “Meta-reinforcement learning of structured exploration strategies,” in *Advances in Neural Information Processing Systems*, 2018, pp. 5302–5311.

[23] T. Xu, Q. Liu, L. Zhao, and J. Peng, “Learning to explore with meta-policy gradient,” *arXiv preprint arXiv:1803.05044*, 2018.

[24] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[25] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in *Advances in neural information processing systems*, 2016, pp. 379–387.

[26] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[27] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *CVPR*, 2017.

[28] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth prediction,” in *ICCV*, October 2019.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[31] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow, “Digging into self-supervised monocular depth estimation,” *arXiv preprint arXiv:1806.01260*, 2018.

[32] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” *arXiv preprint arXiv:1412.7062*, 2014.

[33] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[34] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *ECCV*, 2012.

[35] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” *ECCV*, 2014.

[36] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[37] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[38] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *International conference on machine learning*, 2016, pp. 1928–1937.