Role of reward shaping in object-goal navigation

Srirangan Madhavan
UC San Diego
smadhavan@eng.ucsd.edu

Anwesan Pal
UC San Diego
a2pal@eng.ucsd.edu

Henrik I. Christensen
UC San Diego
hichristensen@eng.ucsd.edu

Abstract

Deep reinforcement learning approaches have been a popular method for visual navigation tasks in the computer vision and robotics community of late. In most cases, the reward function has a binary structure, i.e., a large positive reward is provided when the agent reaches goal state, and a negative step penalty is assigned for every other state in the environment. A sparse signal like this makes the learning process challenging, specially in big environments, where a large number of sequential actions need to be taken to reach the target. We introduce a reward shaping mechanism which gradually adjusts the reward signal based on distance to the goal. Detailed experiments conducted using the AI2-THOR simulation environment demonstrate the efficacy of the proposed approach for object-goal navigation tasks.

1. Introduction

Reward shaping for reinforcement learning is a way to provide localized signals to an agent for encouraging behavior that is consistent with prior knowledge [7]. For the task of indoor robot navigation in search of a target object of interest, it is quite important for an agent to obtain intermediate auxiliary signals based on surrounding objects, to ensure that it's heading towards the goal. This is specially true for large environments, where the the robot may need to take a number of steps to reach the goal [9]. A popular reward function used in the object-goal navigation literature [4,5,11,12,14] is of a binary nature, where a large positive reward is given at the goal state, while a smaller negative step penalty is assigned for every other state. Unfortunately, this type of a signal is quite sparse, thereby discouraging the learning process.

An alternate approach which has gained interest [2, 8] is to use geodesic distance to the closest target as a reward signal. Although this is a denser function compared to the binary reward, absolute knowledge about the closest distance to goal is a strong assumption that may not be easily available outside certain simulation environments [10]. In contrast, we propose a method that relies on the estimated distance to objects calculated via different heuristics. Two approaches which are similar to ours are that of Druon *et al*. [3] and Ye *et al*. [13]. They both provide auxiliary signals based on the bounding box area of objects. However, these rewards are only assigned for the target object, and therefore, the signals are still quite sparse, specially when targets are smaller in size.

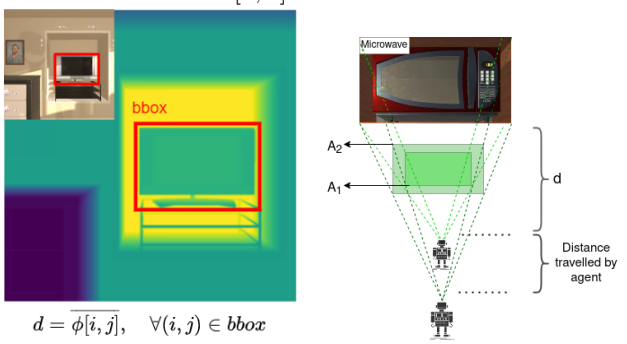
In this work, we build on the initial approach described in [9] by defining distance-based heuristics to modify the reward for both target objects, and other large, salient objects which have a close relationship with the target (called parent objects). In Section 2, we describe two approaches for this. Next, in Section 3, we discuss the results obtained by utilizing the proposed reward shaping mechanism. Finally, we conclude with a discussion in Section 4.

2. Methodology

Pal *et al*. [9] introduced a reward shaping mechanism where the agent receives a "partial" reward, R_p, when it can identify a parent object with close relationship to the target. This is given by $R_p = R_t * Pr(t|p) * k$, where R_t is the target reward, and $Pr(t|p)$ is a probability distribution of the relative "closeness" of all the parent objects, p, to a given target object, t. Additional details can be found in [9]. Additionally, the scaling factor, k, is a constant kept fixed at 0.1. Therefore, the partial reward is independent of the distance between the agent and the parent/target objects, d. Moreover, R_p was only provided when the agent is within a distance threshold from the parent (set as 1m in [9]). In this work, we propose *two* methods to address these problems by reformulating k as a factor of d. Furthermore, we extend the R_p formulation towards both parent, and target objects. The primary motivations for this are: (i) the agent should be encouraged to identify parent objects whenever they are visible, and (ii) by making the reward a factor of d, the agent is further inspired to explore regions closer to p.

(i) Utilizing metric depth - Our first approach involves using metric depth in the form of depth maps obtained directly from the AI2-THOR simulator [6]. In lieu of this, an RGB-D sensor can also be used to get the estimated depth.

From the depth maps, we compute d as the average value of the region, ϕ, bounded by an object's bounding box. This is illustrated in Figure 1a. Subsequently, the scaling factor is formulated as a linear function, $k'(d) = k * (m * d + c)$. In our experiments, $m = -0.15$, and $c = 1$ were empirically chosen to ensure $k' \in [0, 1]$.



(a) Metric distance from depth maps (b) Relative distance from bbox area

Figure 1. Image on the left shows depth map with a bounding box around the object. Inset contains the RGB image of the object. d is obtained by finding the average distance of each pixel in the bounding box. Image on the right shows the relative increase in bounding box area of an object (A_1 to A_2) as the agent moves closer. d is object distance when area is A_2.

(ii) Utilizing bounding box area - While the metric depth approach is intuitive, in theory, we observed that due to the added sensor input in the form of depth maps, the training time increased. Thus, our next approach was to use a heuristic for relative distance, where the scaling factor is calculated based on the assumption that as the agent moves closer, an object's bounding box (bbox) area should proportionately increase. This method, apart from being simple to implement, also reduces the dependence on additional sensor data, thereby minimizing the computational load. For this strategy, the scaling factor is given by $k'(d) = k * (1 - (A_1/A_2(d))^{0.5})$, where A_1 and A_2 are bounding box areas of a particular object in the state, when it was first seen by the agent and the current state respectively. This is depicted in Figure 1b.

In the next section, we validate our proposed hypothesis via extensive experiments.

3. Experiments and Results

We use the AI2-THOR [6] environment for our experiments. The setup and train/test split are consistent with other standard methods - GCN [12], SAVN [11], and MJOLNIR-O/R [9]. We trained the agents for 3×10^6 episodes for each model. Furthermore, for every model, we conduct experiments using 4 different reward functions - binary reward r_{bin}, baseline partial reward from [9], r_{base}, and our two proposed rewards, namely depth-based, r_{depth}, and area-based, r_{area}, respectively. The evaluation metrics adopted from Anderson *et al.* [1].

Metric 1 discussion: Success rate (SR) - Table 1 shows the performance for this metric. For nearly every model, training via the proposed reward mechanism yields the best results, specially for episodes with larger path lengths, *i.e.* $L \geq 5$, where further exploration of the environment might be needed. This shows the benefits of adding a denser reward signal based on distance to objects.

Metric 2 discussion: Success weighted by Path Length (SPL) - As opposed to the results for success rate, the SPL performance drops for the proposed methods. This is shown in Table 2. A possible reason for this could be due to the added incentive that the agent now gets to explore regions around parent objects, before heading towards the target. However, we do not necessarily view this as a major drawback, as exploring the environment is an important feature, specially in large and previously unseen environments.

It should also be noted that generally, the denser distance-based reward functions perform better for models that consider object relationships (like GCN [12], and the MJOLNIRs [9]). This supports our intuition that adding auxiliary signal based on surrounding objects can aid in the search of far-off target objects.

4. Conclusion

We introduced a distance-based reward shaping mechanism that provides a denser feedback to the agent, thereby encouraging it to explore more of the environment. We showed that adopting this strategy leads to higher success rate of reaching the target object for multiple models, specially for cases where the optimal path requires taking a longer sequence of actions. However, due to the added exploration, the path length increases as a result. As part of our future work, we plan to address this issue by adopting imitation learning techniques [4, 5].

Models	$L \geq 1$				$L \geq 5$			
	r_{bin}	r_{base} [9]	ours		r_{bin}	r_{base} [9]	ours	
			r_{depth}	r_{bbox}			r_{depth}	r_{area}
GCN [12]	$33.1_{(0.8)}$	$33.3_{(1.4)}$	$31.7_{(0.7)}$	$\mathbf{35.3_{(0.5)}}$	$25.0_{(1.4)}$	$23.5_{(1.6)}$	$\mathbf{26.9_{(1.1)}}$	$24.6_{(0.8)}$
SAVN [11]	$34.7_{(0.5)}$	$\mathbf{40.7_{(1.4)}}$	$32.2_{(0.9)}$	$39.6_{(0.8)}$	$25.8_{(0.8)}$	$30.0_{(1.4)}$	$26.8_{1.3}$	$\mathbf{31.7_{(1.5)}}$
M_O [9]	$58.8_{(1.0)}$	$64.1_{(0.7)}$	$\mathbf{66.4_{(0.3)}}$	$66.3_{(1)}$	$40.6_{(0.6)}$	$46.6_{(1.6)}$	$50.5_{(0.7)}$	$\mathbf{51.5_{(1.3)}}$
M_R [9]	$65.5_{(0.6)}$	$68_{(0.9)}$	$\mathbf{77.1_{(0.7)}}$	$69.7_{(0.9)}$	$52.3_{(0.8)}$	$52.3_{(0.5)}$	$\mathbf{69.2_{(0.8)}}$	$57.3_{(1.3)}$

Table 1. Metric 1: Success rate (%). The mean score over 5 runs is provided with the standard deviation as sub-scripts.

Models	$L \geq 1$				$L \geq 5$			
	r_{bin}	r_{base} [9]	ours		r_{bin}	r_{base} [9]	ours	
			r_{depth}	r_{bbox}			r_{depth}	r_{area}
GCN [12]	$10.0_{(0.4)}$	$\mathbf{10.8_{(0.5)}}$	$5.5_{(0.2)}$	$8.2_{(0.1)}$	$10.3_{(0.7)}$	$\mathbf{11.2_{0.7}}$	$7.3_{(0.3)}$	$8.7_{(0.3)}$
SAVN [11]	$11.0_{(0.2)}$	$\mathbf{11.1_{(0.3)}}$	$6.6_{(0.3)}$	$10.5_{0.2}$	$11.7_{(0.1)}$	$12.4_{(0.5)}$	$10.5_{0.3}$	$\mathbf{12.8_{(0.6)}}$
M_O [9]	$18.5_{(0.3)}$	$\mathbf{20.7_{(0.2)}}$	$11.6_{(0.1)}$	$15.8_{0.4}$	$17.8_{(0.3)}$	$\mathbf{20.0_{(0.6)}}$	$13.7_{(0.3)}$	$17.3_{(0.5)}$
M_R [9]	$24.4_{(0.3)}$	$\mathbf{26.5_{0.2}}$	$15.0_{(0.3)}$	$16.8_{(0.2)}$	$26.2_{(0.4)}$	$\mathbf{27.2_{(0.3)}}$	$20.3_{(0.4)}$	$19.3_{(0.4)}$

Table 2. Metric 2: SPL (%). The mean score over 5 runs is provided with the standard deviation as sub-scripts.

References

[1] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Roshan Zamir. On evaluation of embodied navigation agents. *CoRR*, abs/1807.06757, 2018. 2

[2] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020. 1

[3] Raphael Druon, Yusuke Yoshiyasu, Asako Kanezaki, and Alassane Watt. Visual object search by learning spatial context. *IEEE Robotics and Automation Letters*, 5(2):1279–1286, 2020. 1

[4] Heming Du, Xin Yu, and Liang Zheng. Learning object relation graph and tentative policy for visual navigation. In *European Conference on Computer Vision*, pages 19–34. Springer, 2020. 1, 2

[5] Heming Du, Xin Yu, and Liang Zheng. {VTN}et: Visual transformer network for object goal navigation. In *International Conference on Learning Representations*, 2021. 1, 2

[6] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474, 2017. 1, 2

[7] Adam Daniel Laud. *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004. 1

[8] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. Thda: Treasure hunt data augmentation for semantic navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15374–15383, October 2021. 1

[9] Anwesan Pal, Yiding Qiu, and Henrik Christensen. Learning hierarchical relationships for object-goal navigation. In *Proceedings of the 2020 Conference on Robot Learning*, Proceedings of Machine Learning Research. PMLR, 16–18 Nov 2021. 1, 2

[10] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1

[11] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to Learn How to Learn: Self-Adaptive Visual Navigation Using Meta-Learning. In *IEEE CVPR*, June 2019. 1, 2

[12] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018. 1, 2

[13] Xin Ye, Zhe Lin, Haoxiang Li, Shibin Zheng, and Yezhou 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)*, pages 6857–6863. IEEE, 2018. 1

[14] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, pages 3357–3364. IEEE, 2017. 1