CS543 Final Project Report

Felipe Arias
University of Illinois, Urbana-Champaign
felipea2@illinois.edu

Victor Gonzalez
University of Illinois, Urbana-Champaign
vag3@illinois.edu

1. Introduction

1.1. Abstract

Visual navigation is the process by which camera-equipped robots find collision free paths to desired locations relying only on their camera input. Despite being an often studied problem, it is difficult for deep learning algorithms to solve due to the size of the state space, partial observability, and the reliability of reinforcement learning algorithms. In this work, we extend popular visual navigation algorithms to include perspectives from two robots rather than one during learning. Usually, the problem is defined from the perspective of the robot that is trying to reach the goal observation (whether it be an explicit image or a semantic description). However, we propose that taking advantage of the camera input of multiple robots could help the learning process due to the additional information a third person perspective provides. In summary, we explore the usage of a third person perspective during visual navigation, propose a new, non-egocentric, goal definition for visual navigation, and show that visual-navigation from a third person perspective is possible in the context of deep reinforcement learning.

1.2. Background

Our work derives from [2], which in turns derives its core structure from [4] and [1].

1.2.1 Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning

The main architecture choices and algorithms used in our work were first proposed in [4]. There, the authors propose a solution for goal-oriented visual navigation as well as AI2THOR, a realistic simulator that we also used in our work. The core of the idea is to use A3C [3], to solve the navigation task by training a function of the target image and observation at the current time step. Figure 1 shows the main diagram from the paper. There you can see that the inputs are the current observation of the robot and the target image that the robot should learn to get to.

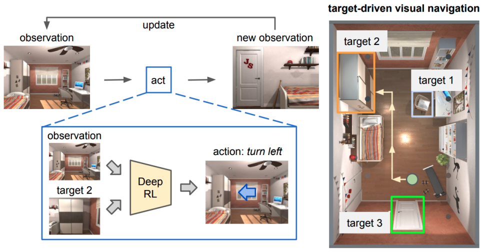

Figure 1. Diagram from [4] showing the core idea and problem formulation.

1.2.2 Reinforcement Learning with Unsupervised Auxiliary Tasks

In order to aid the performance of the reinforcement learning process and enable his own work, the author of [2] utilizes the auxiliary tasks proposed in [1]. The goal of the auxiliary tasks is to provide further regularization on the shared representation that is fed to an LSTM. Specifically, [1] proposes the UNREAL actor as well as pixel control and reward prediction. Which are tasks that attempt to maximize the pixel-wise changes over time and predict the sign of the reward given an action respectively. The network we used in our work included these auxiliary tasks, which can be seen in 2. Additionally, this work proposes the use of a replay buffer, as seen in Figure 2, that contributes well-performing episodes to aid with learning. This replay buffer is also present in our work.

1.2.3 Vision-based Navigation Using Deep Reinforcement Learning

The overall architecture of [2] can be seen in Figure 3. The core contributions of the work that we extended was the addition of two auxiliary tasks. In order to train the convolutional layers faster and have the network pay closer attention to the observation and target images, the author proposed to predict depth and segmentation images of the observations and target image.

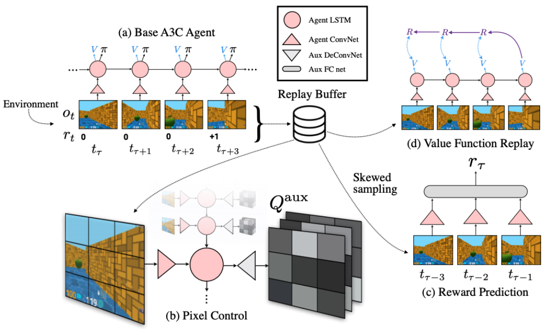

Figure 2. Diagram from [4] showing the core idea, auxiliary tasks, and enhancements.

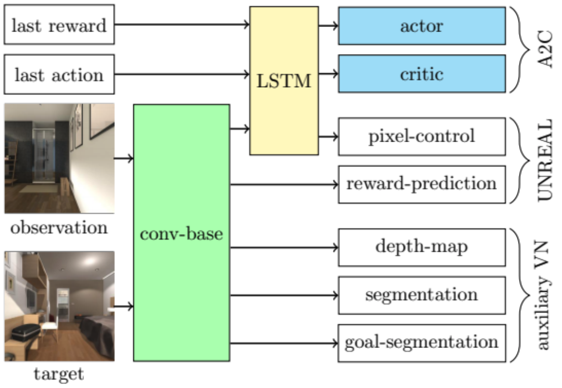

Figure 3. Diagram from [2] showing an overview of the model's architecture.

1.3. Multi-perspective Visual Navigation

Finally, our contribution is the addition and study of visual navigation from a third-person perspective. We added convolutional layers to be able to input images from more than one perspective into the conv-base shown in Figure 3 and studied how the original architecture performed when the inputs are from a third person perspective.

2. Details of the approach

2.1. Data collection

Since our problem differed from the previous work in a fundamental way, our first priority was enabling multiple perspectives within the same environment by placing two robots within it and creating a dataset for our task. Some sample images are shown in Figure 4. As can be seen, each robot configuration is associated with six images, two observations, two segmentation masks, and two depth images; three from a first person perspective and three from a third person perspective.

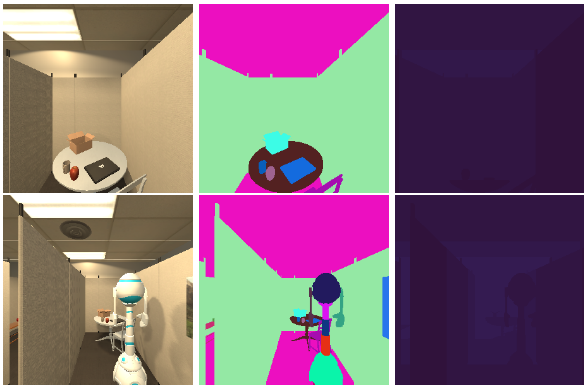

Figure 4. Sample RGB, segmentation, and depth images (respectively). Both views are in the same environment, the bottom row is from the third-person perspective (a second robot) and the top is from the first-person perspective (robot seen in bottom row).

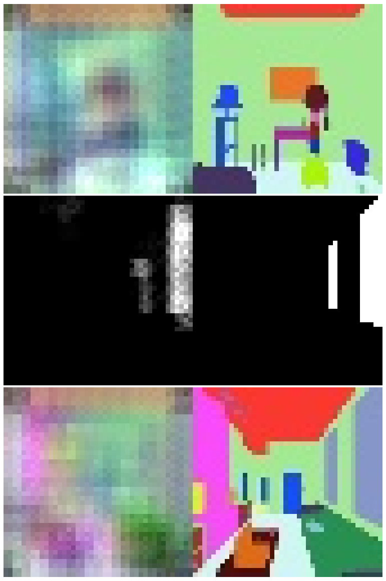

Figure 5. Results pre-training convolutional base on first person perspective images. Top row is goal image segmentation. Middle row is observation image depth prediction. Bottom row is observation image segmentation. The left column corresponds to predictions and right column ground truth.

2.2. Segmentation and Depth Pre-Training

Once we had collected the images from first and third person perspectives, we decided to train the segmentation and depth prediction tasks separately as a way to more easily train the convolutional layers and get practice working with the existing code base. We wrote our own implementation of a training loop and began experimenting.

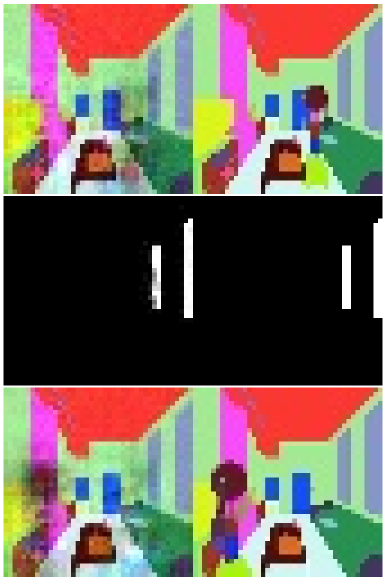

Figure 6. Results pre-training convolutional base on third person perspective images. Top row is goal image segmentation. Middle row is observation image depth prediction. Bottom row is observation image segmentation. The left column corresponds to predictions and right column ground truth.

2.2.1 Segmentation and depth prediction from a first person perspective

Our first task was to pre-train the convolutional base module using observation, segmentation and depth triplets from a first person perspective. This was aimed at recreating the pre-training from [2]. We can see some intermediate results in Figure 5. Further discussion of performance can be found in Section 3.1.

2.2.2 Segmentation and depth prediction from a third person perspective

The next task was to pre-train the convolutional base module using triplets from a third person perspective. This decision came as we knew we would like to train the overall network using only third person images at some point. We can see some intermediate results in Figure 6. Further discussion of performance can be found in Section 3.1.

2.3. Full Training

The challenge of shifting to full training was figuring out how we would add the additional perspectives to the network. In the original convolutional base architecture, seen in Figure 7, there are two pathways that share a base architecture (conv1 and conv2) before being input into a single convolutional section (conv3 and conv4), and then finally passing to a linear layer (fc). The output of the single convolutional section is used for the deconvolutional layers shown in Figure 8 and the output of the linear layer is fed to an LSTM as seen in Figure 3. As such, we would like a way to minimize the disruption to the rest of the network while still allowing input of four total images (target and goal observations, each from first and third person perspective).

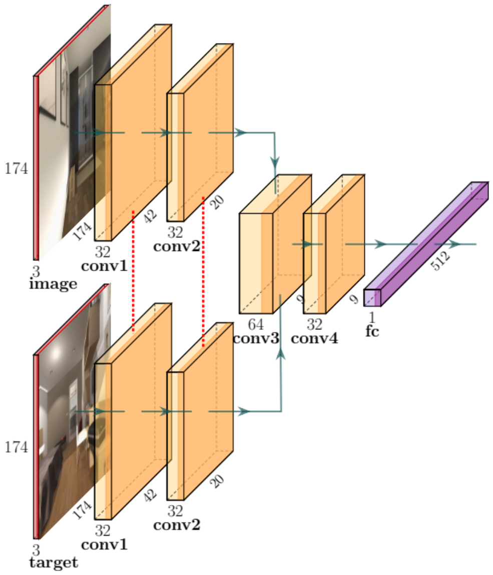

Figure 7. Original convolutional base architecture.

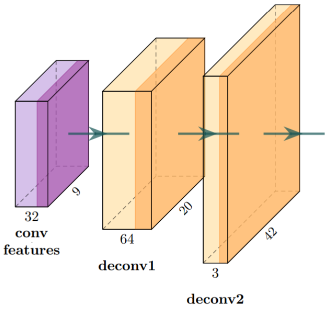

Figure 8. Visual navigation auxiliary task network - observation image segmentation and target segmentation prediction.

Our solution was to expand the concatentation between layers conv2 and conv3. Now, rather than the total input being $20 \times 20 \times 32 * 2$, the total input to conv3 is $20 \times 20 \times 32 * 4$, with the output number layers remaining at 64. The benefit of this solution is that the remainder of the network is unaffected while we still utilize all four images.

3. Results

3.1. Pre-Training

To simulate actual experiments, each batch used a randomly chosen image from outside the batch to treat as the goal image. This was to ensure generalization of our seg-

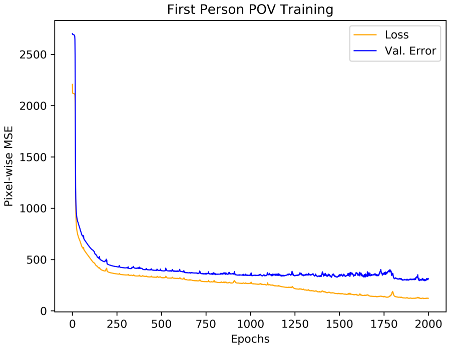

Figure 9. Training plot from pretraining using first person perspective data.

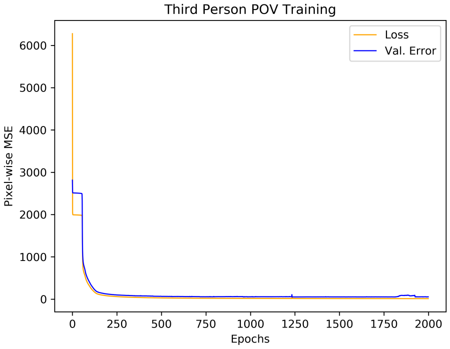

Figure 10. Training plot from pretraining using third person perspective data.

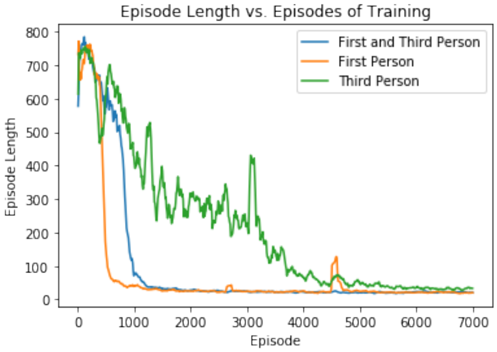

Figure 11. Episode length vs. number of training episodes for single environment experiments

mentation of the goal image. The dataset size comprised of 135 valid robot locations in our discrete robot space, each with 4 different viewing perspectives. In total, this means that we have 540 images. We separate 25 out for usage in validation, meaning we have 515 training images. During validation, we always use the same image as the goal image.

We use the Adam optimizer along with pixel-wise mean squared error for our loss function. We train for 2000 epochs and use our validation images to determine when to stop.

3.1.1 First Person Perspective

Our first person validation images, seen in Figure 5, appear decent, and the segmentation does convey some of the correct info, along with the depth image. Overall, we can see from Figure 9 that the loss reaches 120.4 when the validation error stops improving.

3.1.2 Third Person Perspective

Our third person validation images, seen in Figure 6, appear very good, but this is due to the contents of the images. Our third person camera is fixed throughout training. We can see that the goal image contains the robot on the right-hand side and the observation image contains it on the left-hand side, so the data was generated correctly. Because the background is essentially fixed, the majority of the segmentation looks very good, leading to a MSE loss of 16.18 when validation error stops improving, seen in Figure 10.

3.2. Single Environment Experiments

Our initial experiments conducted in a single environment and with four goal configurations. We had 540 unique configurations, but since for each configuration there is an image from a first person perspective and another from a third person perspective, we had a total of 1080 images, each with its corresponding depth and segmentation mask. As a proof of concept, we ran the original model with a first person perspective, the original model with a third person perspective, and our model with first and third person perspectives. In this subsection we will show the performance of these experiments based on three metrics, how many actions it takes for the robot to get to the goal configuration (episode length), the reward acquired by the robot, and the loss of the auxiliary tasks, which is a combination of the segmentation and depth prediction tasks.

3.3. Episode Length vs. Episodes

As can be seen in 11, the third-person model took considerably longer to converge to the minimal episode length than the first-person model and the combined model. We believe this may be due to the fact that the loss for the third-person perspective is less sensitive to changes in the environment, particularly if the first-person robot is far away, only a few pixels represent the state of the system. Interestingly, although the first person perspective converges faster, the combined perspectives model is more stable and converges to the same value.

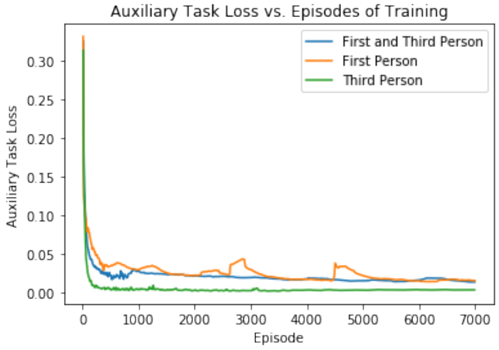

Figure 12. Auxiliary Loss vs. number of training episodes for single environment experiments

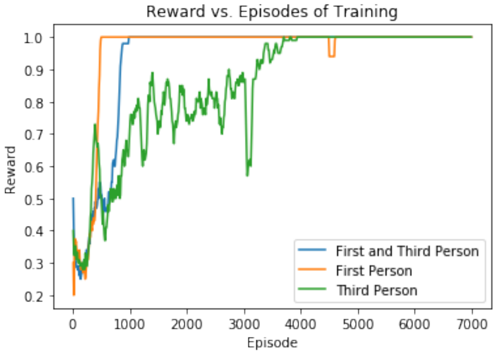

Figure 13. Reward vs. number of training episodes for single environment experiments

3.4. Performance of Visual Segmentation and Depth Prediction

As can be seen in 12, the third-person model converges faster and to a lower auxiliary loss. This is expected since all of the images the third-person perspective sees are from the same viewpoint, the only thing changing over time being the pixels corresponding to the first-person robot. Once again, it can be seen that combined model is more stable and converges to the same value as the first-person model.

3.5. Reward vs. Episodes

As can be seen in 13 and previously mentioned, the third-person model takes considerably longer to converge to the maximal reward. A noteworthy even occurred around episode 4500, where the first-person model was getting suboptimal reward while the combined model maintained the optimal reward throughout.

3.6. Multi-Environment Experiments

Our second set of experiments were conducted in a four different environment, each with a single goal configurations. We had 1968 unique configurations, but since for each configuration there is an image from a first person perspective and another from a third person perspective, we had a total of 3936 images, each with its corresponding depth and segmentation mask. We ran the original model with a first person perspective and our model with both perspectives. Notably, the additional training data and increased state space seemed to improve the performance of the combined model and lead to it performing better than the first-person model.

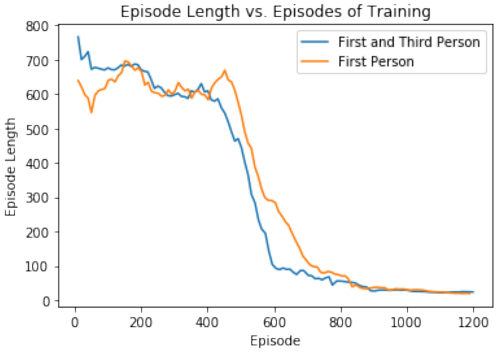

Figure 14. Episode length vs. number of training episodes for multi-environment experiments

3.7. Episode Length vs. Episodes

As can be seen in Figure 14, the combined model outperformed the first person model between episodes 400 and 800, after which both methods converged to the same value. This plot shows that the model is benefiting from the additional perspective and finding shorter paths faster thanks to it.

3.8. Performance of Visual Segmentation and Depth Prediction

In Figure 15, we plot the auxiliary loss of the first person and combined models. Although these were not the only losses that were affected by the additional perspective, we believe that more consistent images across episodes (third-person) allows the network to pay closer attention to the first-person configuration rather than sequences of images necessary to arrive at the goal.

3.9. Reward vs. Episodes

Figure 16 once again shows how the combined model outperforms the first-person model. It is important to note that while both models benefit from the same breakthrough periods of learning, the combined model seems no to loose the insights learned from them like the first person model does. The additional perspective may provide a way to verify what types of sequences of actions are beneficial to learning while there could be more ambiguity only using a first-person perspective.

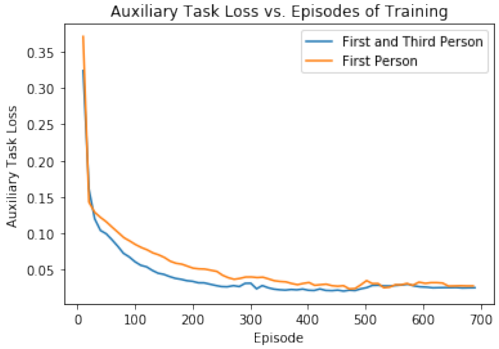

Figure 15. Auxiliary loss vs. number of training episodes for multi-environment experiments

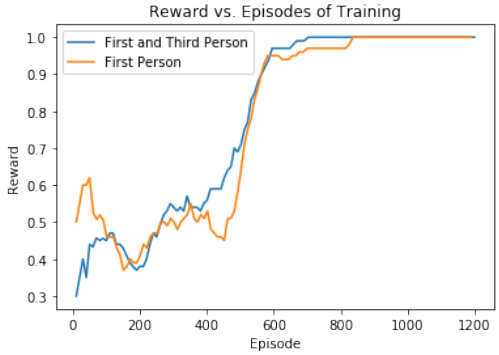

Figure 16. Reward vs. number of training episodes for multi-environment experiments

4. Discussion and conclusions

We are happy with the results we got and would like to continue experimenting with the idea of using multiple perspectives for visual navigation. Most importantly, we showed that the idea can help visual navigation, that it is possible to solve navigation tasks from a third-person perspective (something we had not seen in our preliminary literature review), and proposed a new goal definition for visual navigation (from a third-person perspective).

4.1. Navigation from a Third Person perspective

We studied the advantages and disadvantages of visual navigation from a third person perspective and have concluded that although there are not many use cases other than aiding the navigation of a robot that may or may not have its own camera, it is a worthwhile and understudied task. Our single environment experiments were an attempt to study the three variations on as even footing as possible. Some of the the enhancements to third-person navigation that we have thought of include tracking the first-person robot such that it is always at the center of the image, applying a different loss to an upsampled region of interest (if the goal is far away, too few pixels make the difference between the goal position and one that is not close to it), and allowing the third-person robot to also move and have its actions contribute to the training of the model.

4.2. Enhancing the State of the Art

As we saw in our multi-environment experiments, we have reason to believe that we can improve the existing solutions by using multiple perspectives during training. There are countless architectural improvement we would like to try, from extending the feature vector at the fully connected layer or creating a second one for third-person perspective to only counting the first-person perspective towards the loss, there are many options to improve the current implementation and its performance. We believe, in agreement with the performance changes going from the a single-environment to multiple environments, that additional perspectives would be most beneficial for harder and more diverse problems. Memorizing a sequence of actions that takes you to the goal configuration becomes harder as the number of environments increase, so the third person perspective may allow the network to better estimate where it is in space and reason about its structure.

5. Media and Code

Video and code.

References

[1] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

[2] Jonáš Kulhánek, Erik Derner, Tim de Bruin, and Robert Babuška. Vision-based navigation using deep reinforcement learning. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–8. IEEE, 2019.

[3] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[4] 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 *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.