Creating Dynamic Simulation Environments For Autonomous Navigation

Our machine learning models performed best when trained on datasets created using static textures and the "perfect" navigation path.

Introduction

As Autonomous systems become an increasingly prevalent part of society, developers are turning to machine learning to help these systems make decisions. Since machine learning models are only as effective as the data on which they are trained, roboticists are creating datasets in virtual environments as they are safer, more dynamic, and cost effective. However, simulation often comes at a cost--models created using synthetic data perform differently in simulation and reality. In order to bridge the gap between simulation and reality, we employ a variety of features in our hyper-realistic environment that we created using Unreal Engine 5 (UE5) . These features include varied lighting, texture, and randomization, as well as dynamic movement to allow for a diverse and effective dataset.

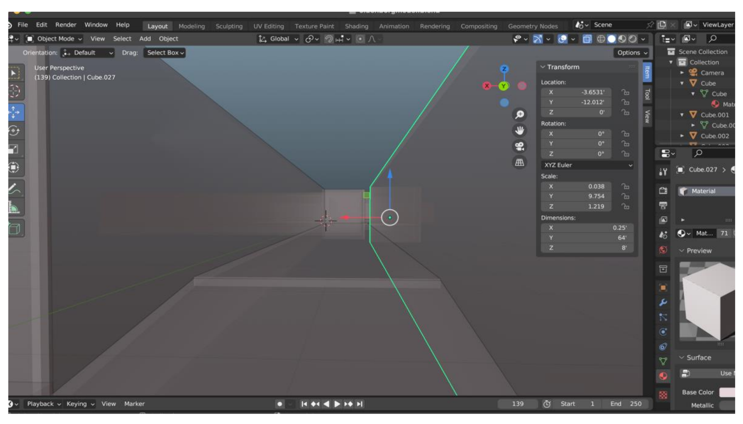

Fig. 3: Oldenborg hallway modeled in Blender

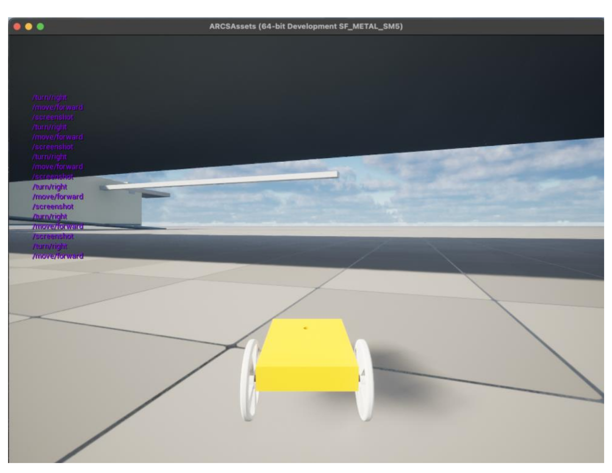

Fig. 4: Agent in UE5 receiving commands

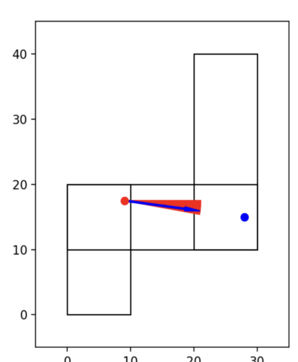

Fig. 5: Agent navigates around some directed corridors represented as overlapping boxes for data collection.

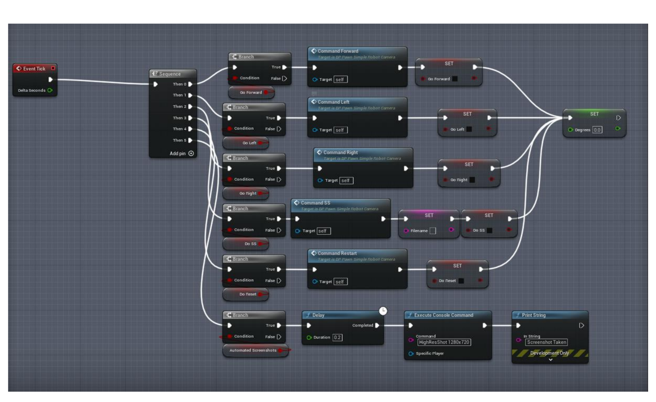

Fig. 6: Exploration with Open Sound Control

Methods

Data Collection:

Figure 1 shows real-world Oldenborg next to the corresponding UE5 environment with randomized colors and textures. We constructed the simulation environment by modeling Oldenborg in Blender and then importing it into UE5. The simulation enables us to dynamically change wall, ceiling, and floor textures.

We created several agents to navigate the simulation and collect data by following different paths. For example, a "perfect" navigator will follow a path down the center of each hallway and turn 90 degrees at each corner, whereas a "wandering" navigator will follow a similar path but will occasionally take random actions making it deviate from the perfect path and collect data from more varied angles.

To compare these features, we collected four datasets by pairing (1) perfect navigation and static textures, (2) perfect navigation and randomized textures, (3) wandering navigation and static textures, and (4) wandering navigation and randomized textures.

Training:

We trained a ResNet-18 model using fastai on the datasets with the task of navigating from one point in the simulation environment to another. Additionally, we implemented an image + command technique that takes the current image combined with the previous output action to form the input that is used to predict the next action (left, right, forward). We explored using both pre-trained and non-pre-trained models and found that using pre-trained models drastically improved performance, likely due to the small dataset size.

Testing:

In order to evaluate the model's performance, we assess its ability to predict actions (left, right, or forward) that lead to the desired outcome in our simulation environment. Correct actions are computed using a map of the environment that is not available to the robot while training models.

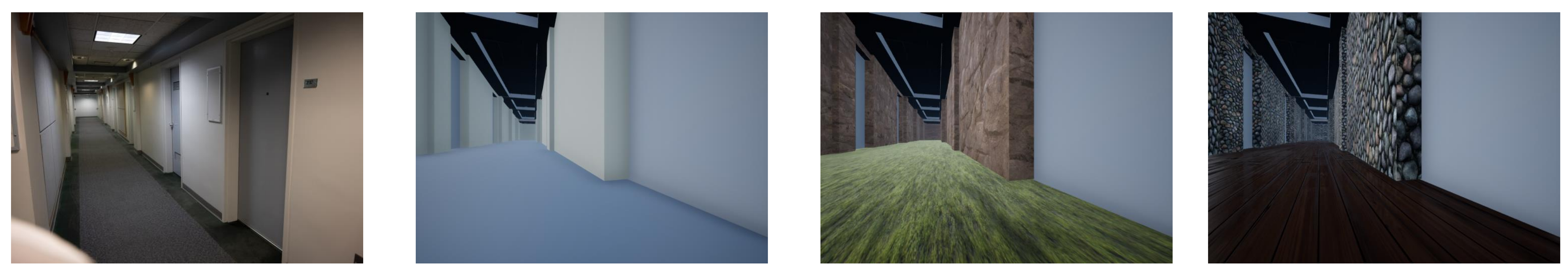

Fig. 1: (Top-Left) Photograph of a hallway in Oldenborg Hall on Pomona College's campus. All other images depict the same hallway in simulation with different randomly selected textures.

Results

Our evaluation indicates that Dataset (1) created using perfect navigation and static textures achieves the best performance in the static environment and the worst performance in the dynamic environment. On the other hand, while models trained using Dataset (4) which was created using wandering navigation and random textures have the lowest accuracy, they do have the best performance in the environment with textures that change randomly. Furthermore, although using a pretrained network improves performance, it is important for our next experiments to include larger datasets with more variability–both in textures and in the viewpoints represented in the captured images.

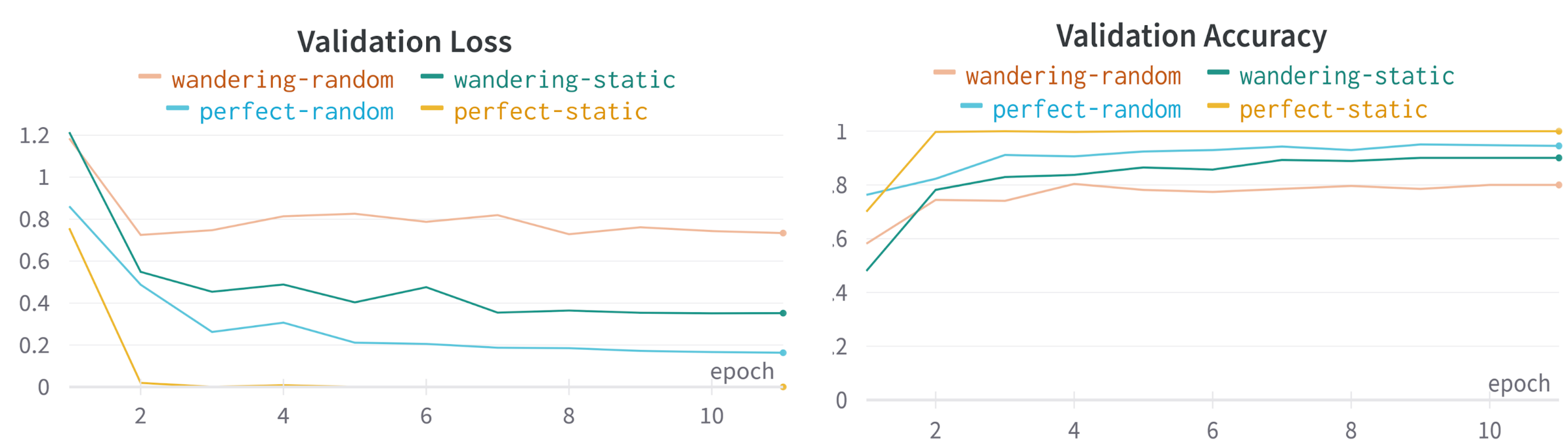

Fig. 2: Validation accuracy and loss for the models trained using different datasets.

Conclusion

Synthetic data will play a major part in the future of robotics. Reconstruction techniques such as photogrammetry, AI-generated content (AIGC) like stable diffusion, and more conventional procedural generation algorithms will be combined to generate data that is both realistic and challenging. Models trained with these environments will have a better chance of surviving the transition from a virtual space into the real world.

In our future work, we will be primarily continuing in two directions. First, we will explore additional randomized features, such as lighting and dynamic actors. Second, we will evaluate trained models in the real world on our prototype robot.

Collaborators

Daisy Abbott, Liz Johnson, Christy Marchese, Anjali Nuggehalli, Francisco Morales Puente, Chau Vu, Ella Zhu, Prof. Anthony Clark

For more information, please scan our QR code for a PDF abstract of our paper.

Scan me!

Pomona College 1887