Self-Supervised, Goal-Conditioned Policies for Navigation in Unstructured Environments

Travis Manderson*, Juan Camilo Gamboa*, Stefan Wapnick*, Jean-François Tremblay*, Hanqing Zhao*, Florian Shkurti†, Dave Meger* and Gregory Dudek*

*Mobile Robotics Laboratory, School of Computer Science, McGill University, Montreal, Canada
†Robot Vision & Learning Lab, Department of Computer Science, University of Toronto, Canada

Email: {travism,gamboa,swapnick,jft,_hzhaod,bmeger,dudek}@cim.mcgill.ca, florian@cs.toronto.edu

Abstract—We present a goal-conditioned visual navigation system trained using hindsight relabelling and self-supervision that learns a control policy for close-proximity robot inspection in natural, rugged environments. Our technique enables robots to navigate collision free to sparse geographic waypoints provided by the user without any prior map of the environment.

We first learn a safe policy that greedily navigates to examine regions of interest while avoiding collisions but does not take into account specific geometric goals. We then augment this policy via the addition of goal-conditioning to seek our specific waypoints and a final goal.

The first policy synthesis is achieved through either behavior cloning or self-supervised learning. This policy is extended to be goal-conditioned, using hindsight labelling, guided by the robot’s relative localization system without any manual annotation. The goal-conditioned policy can then be deployed to navigate a series of arbitrary goals (waypoints).

We validated our approach on an underwater vehicle in a difficult open ocean environment to collect scientifically relevant data on coral reefs while operating safely and autonomously at an impressive proximity of 0.5 m to the coral. We also demonstrate this approach to terrestrial navigation, illustrated using a 1:5 scale, off-road vehicle.

I. INTRODUCTION

In this paper, we present a visual navigation system that learns safe, reactive behaviours to make close-proximity robot inspections in rugged and challenging environments. Our approach uses a combination of self-supervised reinforcement learning and hindsight labelling to synthesize collision avoidance, informative path planning, and goal-directed navigation into a single policy that enables a robot to reach relative geometric waypoints provided by a user without any prior map. Our system is generic enough to be used in multiple unstructured domains, for example, underwater or on-ground, to autonomously collect scientific data critical for environmental monitoring and understanding.

Our approach begins by learning a non-goal-conditioned but safe navigation policy that seeks preferable terrain characteristics or scientifically desirable observations. In our previous work, we have shown that this policy can be trained via self-supervised reinforcement learning using a hybrid model-based and model-free network [25] or through behavior cloning where a relatively small set of images are labeled with steering commands that would direct the robot towards regions of interest [23] [24]. Using these policies, the robot is able to navigate safely following desired behaviour, but it is unable to reach specific geometric points.

We address this shortcoming in our proposed method by augmenting this policy with goal-seeking behaviour using hindsight experience relabelling [11]. We generate a dataset of images, actions, and position (as measured by a vision-based state estimator running onboard) collected from exploration experience using the non-goal-conditioned policy. We split the recorded trajectories into sub-segments and interpret the end position of each sub-segment as a goal waypoint. In this way, each segment is automatically labelled for goal-directed navigation from the beginning to end. We then use these goal-directed segments to train a goal-conditioned navigation policy that achieves safe, well behaved, and goal-directed navigation.

We have demonstrated our system in an underwater ocean environment with difficult visual conditions and external disturbances (Fig. 1 left), as well as performed preliminary validation on a 1:5 scale off-road vehicle (Fig. 1 right).

II. BACKGROUND AND RELATED WORK

Our work builds upon existing literature on sensor-based navigation, imitation learning, and reinforcement learning for autonomous robot navigation in the field. Natural and unstructured environments, such as underwater or forests, often present unique navigation challenges, including incomplete knowledge of the surroundings, reacting to environmental disturbances, identifying free space, and inferring traversable terrain.

A common visual navigation approach in these environments is to first perform semantic segmentation followed by geometric path planning [10] [38] [41] [43]. Even with segmentation, often assigning a weight corresponding to the
traversability of each segmentation class is difficult. One approach is to infer the traversable classes using expert demonstrations. Another technique of relying on expert demonstrations is to directly predict actions from input images. Simple imitation can be done via supervised learning (behavioural cloning) on observation and action pairs from an expert, however, it is not robust to distribution shift. Several authors have proposed data augmentation or domain matching methods for increasing robustness, but the underlying problem of compounding errors during distribution shift remains. DAgger addresses this problem by iteratively querying the expert on states that are visited by the learned policy during evaluation. In addition to ground vehicles, this technique has also been demonstrated on both aerial vehicles and underwater robot systems. More recent methods have combined human demonstrations and imitation learning with end-to-end reinforcement learning as another way to handle distribution shift.

Several authors have also used imitation learning as a way of identifying natural paths in the environment. Demonstrations are often collected with one camera directed directly towards the path and several other cameras mounted with offset positions. In this way, the images captured can be automatically labelled with the corresponding action. A robot should take to steer towards the path. The goal-conditioned model can be successfully used to simultaneously traverse smooth terrain, avoid obstacles, and reach a set of desired waypoints.

**Imitation Learning:** Another technique of relying on expert demonstrations is to directly predict actions from input images. Simple imitation can be done via supervised learning (behavioural cloning) on observation and action pairs from an expert; however, it is not robust to distribution shift. Several authors have proposed data augmentation or domain matching methods for increasing robustness, but the underlying problem of compounding errors during distribution shift remains. DAgger addresses this problem by iteratively querying the expert on states that are visited by the learned policy during evaluation. In addition to ground vehicles, this technique has also been demonstrated on both aerial vehicles and underwater robot systems. More recent methods have combined human demonstrations and imitation learning with end-to-end reinforcement learning as another way to handle distribution shift.

Several authors have also used imitation learning as a way of identifying natural paths in the environment. Demonstrations are often collected with one camera directed directly towards the path and several other cameras mounted with offset positions. In this way, the images captured can be automatically labelled with the corresponding action a robot should take to steer towards the path. The authors assert that VPNs can be viewed as jointly model-based and model-free and found them to be more sample efficient and to generalize better than Deep Q-Networks.

Khan et al. developed a similar architecture, known as Generalized Computation Graphs (GCGs), for the task of predicting obstacles for a short horizon and demonstrated it on a remote-controlled car in an indoor hallway environment.

**Goal-Conditioned Navigation:** While typical navigation learning scenarios consider a single task, such as lane following or navigating towards a fixed goal, we want also to consider goal-conditioned policies. Conditional imitation learning has been studied extensively in the last few years and has many connections to goal-conditioned reinforcement learning, particularly in the batch case.

**Self-Supervised Learning:** The idea of learning terrain classes or navigation strategies directly from sensor data has been examined by several authors. Giguere et al. and looked at learning terrain classification and gait policy selection directly from unsupervised or minimally-labelled Inertial Measurement Unit (IMU) data. Wellhausen et al. used recorded force-torque signals from a quadruped robot to label traversed terrain. This data was projected into the robot’s camera frame for generating a training set for segmentation.

Our work is close in motivation to informative path planning, where most methods need to estimate the surrounding map while executing frontier-based exploration. In contrast, our work does not assume a map because it does not need to perform exploration exploitation within a map due to the direct use of reactive policies instead of a pipeline consisting of perception, mapping, path planning, and tracking.

**III. Approach**

Our system is designed with the purpose of extracting a goal-conditioned policy, useful for navigation, from a lower level policy trained via self-supervised learning. The purpose of our design is that high-level behaviours, such as relevant scientific data collection or waypoint following, is harmoniously built on top of low-level behaviors like obstacle avoidance. Although our method is flexible enough to work with behavioural
cloned policies, for the purposes of this section, we focus on the model used for self-supervised learning for off-road navigation. Details on our goal-conditioned behavioral model can be found in [24].

A. Model Overview

We first start with a non-goal-conditioned deep learning model for predicting the terrain roughness and collision probabilities over a fixed horizon. Our architecture is similar to Khan et al. [17], which was previously shown to produce good performance for short-term control situations. The model architecture, as exemplified in Fig. 2 operates on the recent visual history of the past four images, representing the image state $s_t$ and an action sequence of steering commands $⟨a_t, a_{t+1}, ... a_{t+H-1}⟩$. The image state is passed through a convolution layer to form the initial hidden state $h_t$ of the Long Short-Term Memory (LSTM).

At each timestep, the model predicts the probability of each terrain class over the planning horizon $(H)$: $p(y_{t+1}, y_{t+2}, ..., y_{t+H})$, where $y_t \in C$ and $C$ is the set of all terrain classes. We choose increasing label values to be rougher terrain (0 being completely smooth and $|C|−1$ being an obstacle). This architecture models the joint probability of terrain classes over the horizon while assuming conditional independence between labels:

$$p(y_{t+1}, ..., y_{t+H} | a_{t+H−1}...a_t, s_t) = \prod_{i=1}^{H} p(y_{t+i} | a_{t+i−1}...a_t, s_t)$$

The cross-entropy loss is used to train the model with L2 regularization:

$$L_t = - \sum_{i=1}^{H} \sum_{c_j \in C} y_{t+i} = c_j \log p(\hat{y}_{t+i} | a_{t+i−1}...a_t, s_t) + \lambda \|w\|_2^2$$

where $y_{t+i}$ and $\hat{y}_{t+i}$ are the true and predicted labels.

B. Planning

Planning is performed by maximizing the expected reward over the planning horizon. Each terrain class is assigned a reward, with smooth terrain having the highest reward and the obstacle class having a reward of zero:

$$A^H_t = \arg \max_{a_{t+1}...a_{t+H−1}} - \sum_{i=1}^{H} \sum_{c_j \in C} c_j * p(\hat{y}_{t+i} = c_j | a_{t+i−1}...a_t, s_t)$$

A randomized K-shooting method is used to generate a random set of K action rollouts (trajectory) at each timestep: $A^H_{(t,k)} = (a_{t+1}...a_{t+H−1})_k$. The expected cumulative reward for each trajectory is the sum of the individual rewards for each predicted terrain class. Our policy applies the first action from the trajectory with the highest perceived reward. Planning is repeated at every timestep, in an MPC fashion, to compensate for modelling errors.

C. System Overview

Our experimental vehicle used for off-road driving is a 1:5 scale RC buggy, as shown in Fig. 3. An embedded microcontroller is used to control a drive motor and two servo motors are used for steering and braking. The microcontroller also performs pose estimation using a Kalman filter to fuse information from several redundant IMUs and an RTK GPS. An Intel i7 NUC running Ubuntu and Robot Operating System (ROS) records camera images, Lidar obstacle detections, and IMU readings. An NVIDIA Jetson Xavier, also running ROS, runs our deep learning model implemented in Tensorflow [11]. The Lidar is used only to perform short-range collision detection and act as a bump sensor and measure the instantaneous presence of an obstacle.

The IMU is used to measure terrain roughness, which has previously been considered by other authors [5,13,45]. We measure the Root Mean Square (RMS) linear acceleration reported by the robot’s IMU over short time windows of 20 samples collected at 60 Hz to approximate an instantaneous roughness score of the traversed terrain.

D. Hindsight Relabelling

To train the goal-conditioned model, we generate a location-aware dataset by allowing the robot to explore using the non-goal-conditioned terrain model while recording the relative pose onboard. These trajectories are broken up into examples used for the navigation task: which action should the robot execute to arrive at a particular location?

Goals are sampled from the collected trajectories in a method similar to [2]. For each training mini-batch, a random set of trajectories and timesteps within those trajectories are selected from the dataset. Each sampled item is randomly associated with a future timestep (limited to ensure the goal is not excessively far away), which is interpreted as a goal. After association with a goal from hindsight relabelling, training tuples that can be applied to the goal-conditioned network are in the format $⟨s_t, g_t, \hat{a}_t⟩$, where $s_t$ is the image state, $g_t$ is the relative goal, and $\hat{a}_t$ is the action taken.

E. Goal Conditioning

Our proposed goal-conditioned navigation policy is implemented by augmenting the non-goal-conditioned terrain prediction model with an input waypoint $g_t$ relative to the robot’s current frame. The input vector is passed through a fully-connected layer before being multiplied with the output of the convolution layer. In practice, we assume our method can be used when there is no access to global position coordinates. Therefore, we use onboard state estimation (e.g. IMU, GPS or visual odometry) to compute the relative offset to the waypoints that are with respect to the robot’s starting position.
The model objective function is modified to predict the distance between the input action sequence \(\langle a_t, a_{t+1}, ..., a_{t+H-1}\rangle\) and the record action sequence \(\langle \hat{a}_t, \hat{a}_{t+1}, ..., \hat{a}_{t+H-1}\rangle\). In this way, we reward predictions with the lowest distance. We perform planning in a similar fashion to the non-goal-conditioned model by choosing the trajectory with the highest perceived cumulative reward and take the first action in the sequence.

IV. EXPERIMENTS

Off-road Driving: We validated the self-supervised learning of terrain classes and the non-goal-conditioned model in the real-world, as shown in Fig. 4 on the 1:5 scale RC buggy described in section III-C. Despite a relatively small dataset, we obtained reasonable performance after \(\approx 8,000\) training steps. The prediction accuracy ranged from 78% to 60% over horizon lengths of 1 to 16 respectively. We also performed a quantitative evaluation of our model in simulation for four terrain classes: smooth, medium, rough, and collision. As shown in Table I on-policy navigation significantly reduced the amount of rough terrain encountered from 42% to 11% and drove on smooth terrain 80% of the time.

<table>
<thead>
<tr>
<th>Percentage of Terrain Traversed in Simulation While Driving on-Policy and Random.</th>
</tr>
</thead>
<tbody>
<tr>
<td>smooth</td>
</tr>
<tr>
<td>on-policy</td>
</tr>
<tr>
<td>random</td>
</tr>
</tbody>
</table>

We performed a preliminary validation of our goal-conditioned model in simulation for off-road driving. We created training trajectories and sampled successive waypoints from a normal distribution with a mean of \(\approx 28\) m and a standard derivation of \(\approx 13\) m.

Fig. 5 illustrates the predicted action sequence from the goal-conditioned model that guides the robot towards the goal while still inheriting the obstacle avoidance and smooth terrain preferring behavior from the non-goal-conditioned model. Fig. 6 shows an example trajectory traversed by the robot to reach a set of 8 waypoints while maintaining these characteristics.

Underwater Navigation: We have also successfully deployed a variant of the goal-conditioned policy in the open ocean on the underwater vehicle shown in Fig. 1 (left). In these experiments, we initially trained a behavioural imitation model described in [24] to avoid obstacles and prefer swimming over coral reef for scientific data sampling. Fig. 7 shows one trajectory that was executed to reach a set of waypoints. Overall, we ran trials executing 4 different 10-waypoint trajectories collectively spanning over 1 km. One of these trajectories reached 8 out of 10 waypoints, two reached 7 out of 10 waypoints, and one reached all 10 waypoints.

V. CONCLUSION

In this paper, we have described a self-supervised approach to learn goal-conditioned navigation policies in unstructured and natural environments. Additionally, we have examined its application in both off-road driving in rugged environments and in demanding underwater environments where the presence of surge, weather variations, and lighting variations cannot be overstated.
REFERENCES


2018.


