

Chapter 1

Methodology

We develop a framework to learn and predict traversability that learns directly on ground patches for any robot using purely simulated data. The main idea is to run a robot into a simulator on different synthetic terrain and later used its interactions to train an estimator. Simulation data has two main advantages: varirity, cost and speed. With simulations, we can easily increase the number of data samples by just include more maps, while doing the same with a real robot requires to first identify the suitable ground and then travel to it. Clearly, this also decrease the time required to collect the robot's interaction with the ground. The time could be additional decreased by running different simulations in paralla avoiding to physical manufactor or buy additional robots. Moreover, we are not limited by physical constrain and therefore the robot can interact with a multitude of different grounds. For instance, to collected real world data the robot must be physically transport to different locations, on the other hand, with simulations we can immediately load a new map.

To create a quality dataset, we must first generate a series of various surfaces and ensure their variority. Intuitively, those maps should be small enough to allow a quick interaction with their features but big enough to include variority. With modern ground generation techiques it is fairly straightforward to generate a rich array of terrains with different characteristics such us bumps, ramps, slopes in different levels and sizes.

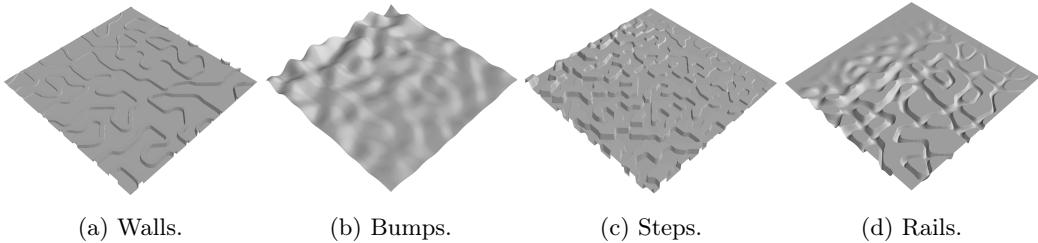


Figure 1.1. Some artificially generated terrains, $10 \times 10\text{m}$, with one specific feature each.

This maps can be loaded into a simulator to let the robot interact with them. To ensure a correct exploration of each terrain, it is better to randomly spawn the robot multiple times

on the same map and let it walk for a fixed amount of time. The robot position is tracked and stored during simulation in order to extract a small portion of ground, patches, around each of its trajectory's position. Those patches are label using a minimum advancement defined for the robot and then directly feed to a deep neural network.

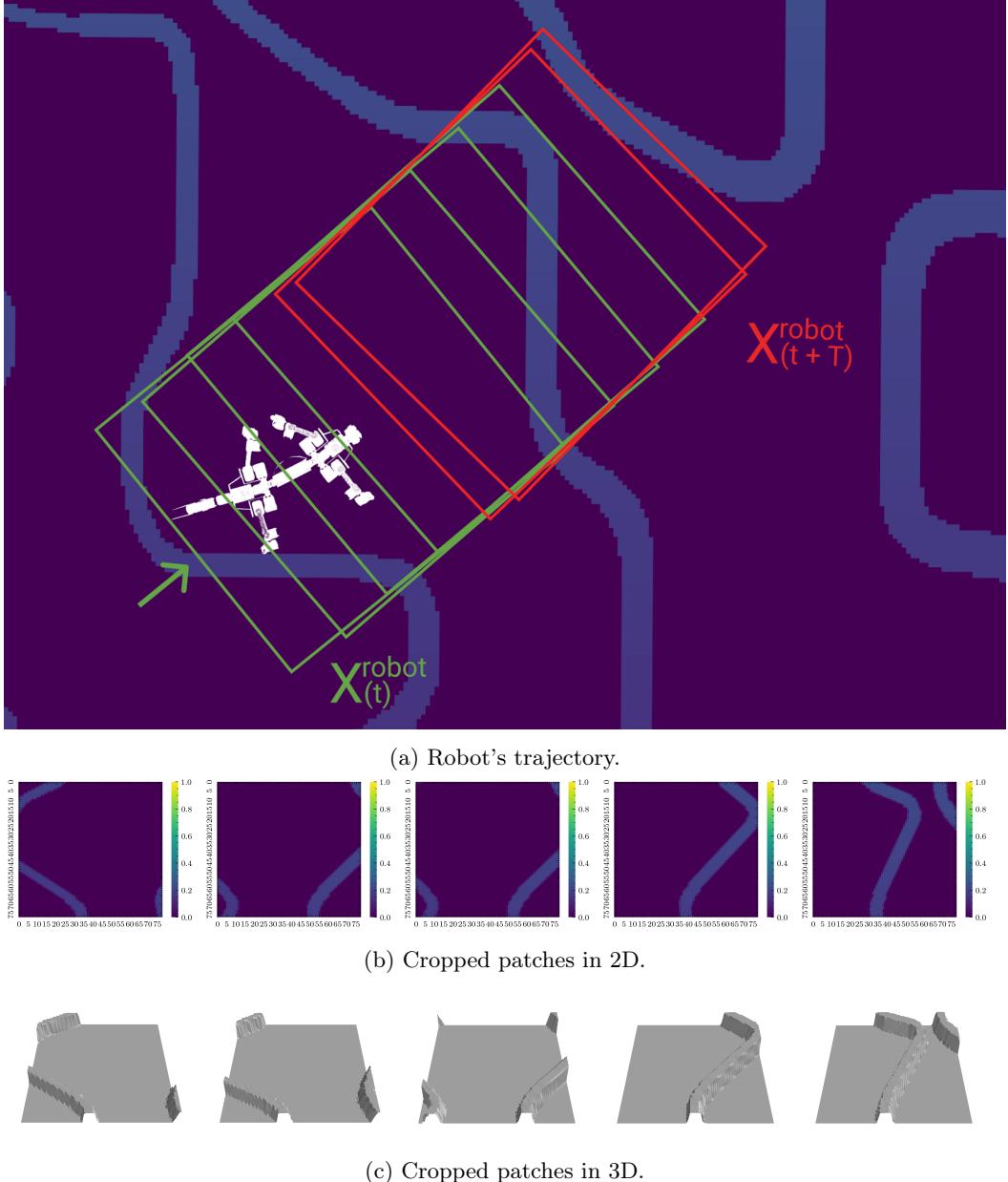


Figure 1.2. Example of a robot trajectory (a) extracted during training on a map with different walls. Robot's initial position is showed by its white silhouette. Patches borders are label if traversable and red if not and showed as 2D (b) and 3D(c) rendered images.

We used deep learning techniques to directly lean features from raw data, in our case images, without needing to preprocess the inputs as it was done historically.

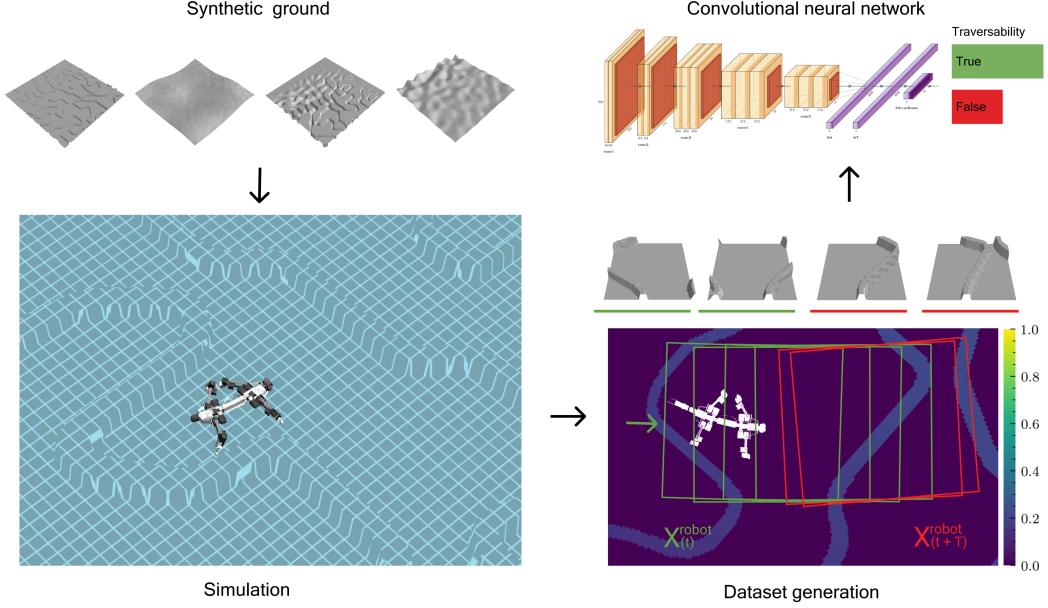


Figure 1.3. The proposed traversability framework’s main blocks in counter-clockwise order. First we generated meaningful synthetic ground, then we let the robot spawn and walk on them in simulated environment while storing its interactions. Later, we crop a region of ground, a patch, for each simulation trajectory around the robot according to its locomotion. We label those images using a defined threshold and fit a deep convolutional neural network to predict traversability.

