Sim-to-Real: Virtual Guidance for Robot Navigation

Evan Luo*, Yu-Wen Chen*, Kai-Chen Lin*, Chao-Hsien Ting*, Hao-Kang Liu*, Chu-Chi Chiu*, Yu-Wei Chu*, Chien Liu, Hsin-Wei Hsiao, and Chun-Yi Lee

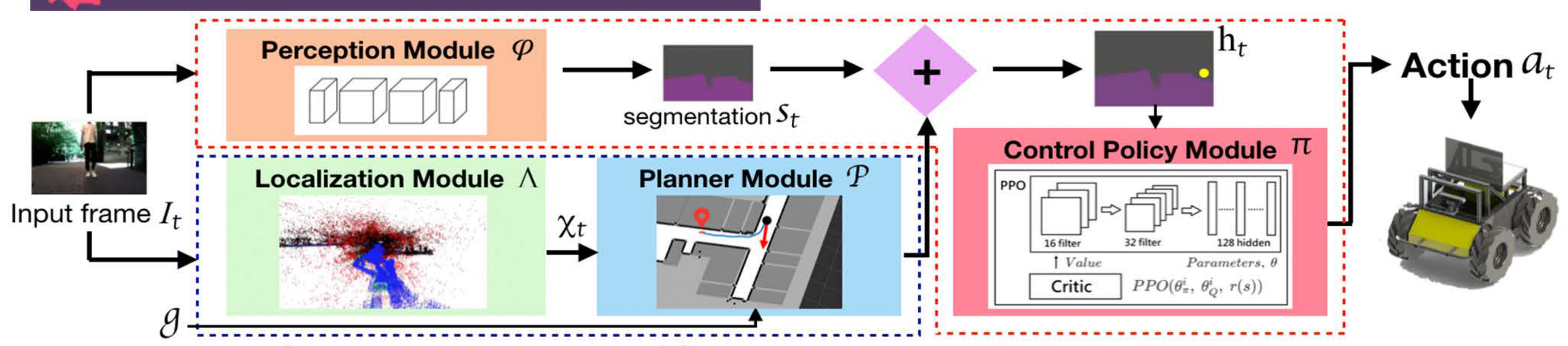
Elsa Lab, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

Abstract

We present an effective, low-cost, and easy-to-implement modular framework for completing complex navigation tasks. Our proposed method is based on a single monocular camera to localize, plan, and navigate. A localization module in our framework first localizes and acquires the robot's pose, which is then forwarded to our planner module to generate a global path and its intermediate waypoints. This information along with the pose of the robot is then reinterpreted by our framework to form the "virtualguide", which serves as a virtual lure for enticing the robot to move toward a specific direction. We evaluate our framework on a Husky robot in a number of virtual and real-world environments, and validate that our framework is able to adapt to unfamiliar environments and demonstrate robustness to various environmental conditions.

Proposed Methodology

Overview of framework



The proposed framework consists of four modules:
a localization module, planner module, perception module, and control policy module.

- The localization module is responsible for estimating the current pose of the robot from the visual input, and conveys it to the planner.
- The planner module constructs a path between the robot's current location and the desired goal, defining a global direction for the robot to follow.
- The perception module translates RGB images from the monocular cameras into scene semantic segmentation.
- This global direction is then reinterpreted by our framework as the tendency, which is rendered as the virtual guide on the semantic segmentation input view. The virtual guide scheme in the real world aims at rendering a virtual guide at suitable locations on segmented input such that the robot is able to follow the path derived by planner module and navigate to the goal.
- The control policy module is implemented as a DRL agent, with an aim to learn a policy for chasing the virtual guide and avoiding obstacle collision.

Training methodology

- Our robot uses only a single monocular camera for navigation, without assuming any usage of LIDAR, stereo camera, or odometry information from the robot.
- During training, the control module only receives the image segmentations rendered by simulators.
- During execution, the control module receives image segmentations from the perception module.
- The perception module is only used during execution. It can be any semantic segmentation model.
- The virtual guide is depicted as a yellow ball in this work, however, it is not restricted to any specific form of representation. The location is selected based on global direction and whether obstacles exist in relevant areas.

(a) The robot's pose. (b) The virtual guide.

Training environments

- Our control policy model is trained using twenty independent maps.
- A variety of obstacles are placed to resemble more to real world scenarios, bridging the gap between simulation and reality.
- The size and field of view of the agent is the same as our robot agent Husky and monocular camera view mounted on it.
- At the beginning of each episode, a map is randomly selected, with the agent and the goal placed at randomly determined locations.
- Our reward function is designed such that a one-time reward is awarded to the agent if the distance between the virtual guide and the agent is within a threshold.
- No time penalty is applied during training.

Illustration of 20 training maps

Experimental Results

Model settings and Robotic Platforms

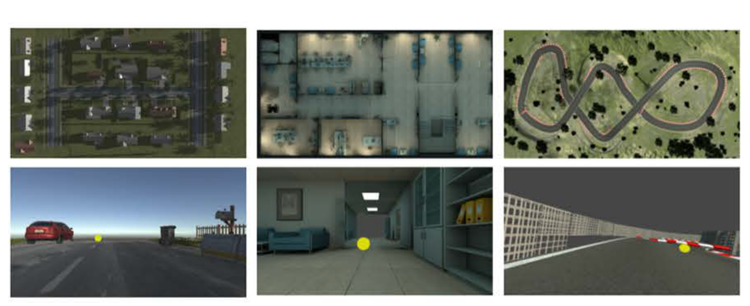
Control policy module
SimpleNet with 3 fully connected layers and 128 hidden units.
Segmentation model
Retrained Google's MobileNetv2 on Cityscapes, ADE20K, and our self-obtained dataset.
Localization module
ORB-SLAM2 with a maximum of 1500 features per image.
Planner module
Dijkstra's path planning algorithm.

(a) Husky CAD model. (b) Outdoor navigation.

Clearpath Husky Autonomous Ground Vehicle (AGV) for both outdoor and indoor navigation.
Run on a laptop with Intel i7-9750H CPU and NVIDIA GTX 1660Ti GPU 16G RAM.

Virtual evaluation

Three different virtual environments :
A suburban area, a dimly lit office, and a race track.
Achievements :
At least 75% success rates in each virtual environment, validating that our model is able to act accordingly with our virtual guidance scheme in unfamiliar environments.



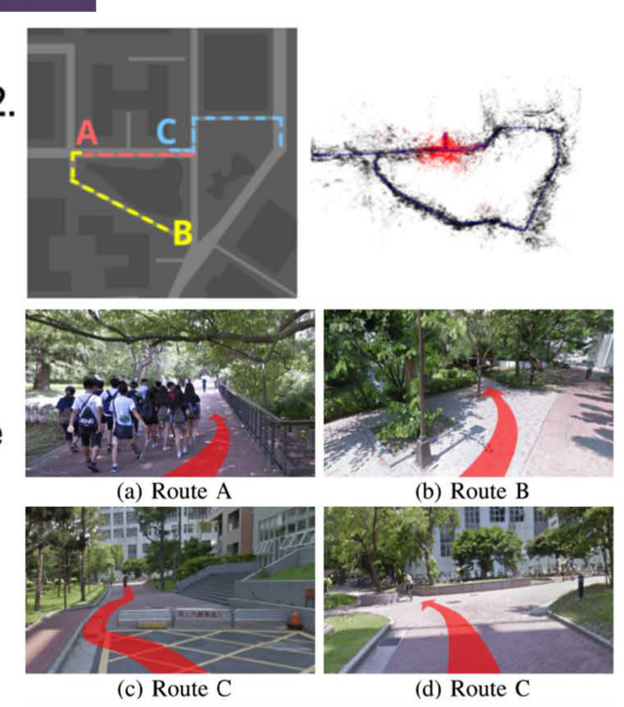
(a) Suburban (b) Office (c) Race track

Environment	Success Rate (%)	Duration(s) μ	Duration(s) σ
Suburban	76.56	405.4	7.486
Race track	92	335.8	2.39
Office	94	176.1	2.02

Outdoor experiments

Three diverse routes :
- Maps are pre-built by ORB-SLAM2.
- Environments include rugged terrains and sidewalk-road transitions and large obstacles not visible in pre-built maps.

Achievements :
- Achieves an average success rate of 83% on route A,B, and C.
- The control policy module is the same for both indoor and outdoor evaluation tasks.
- Results demonstrate our system's ability to navigate through outdoor diverse terrain.

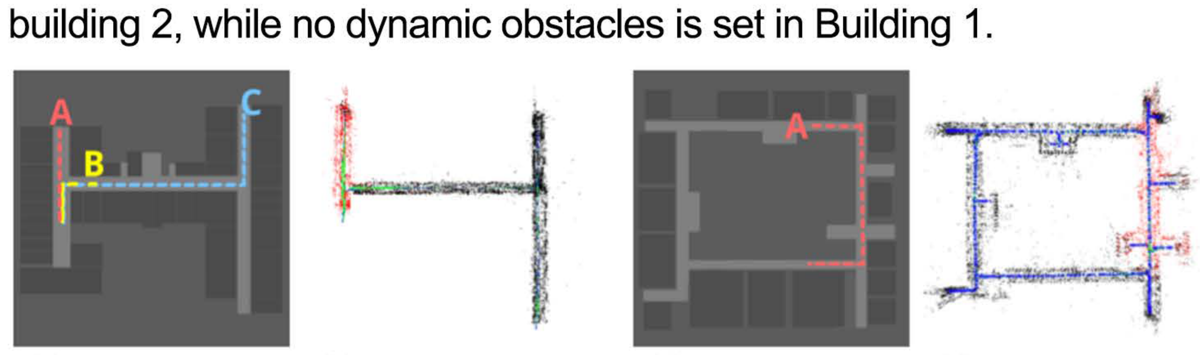


(a) Route A (b) Route B (c) Route C (d) Route C

Environment	Length (m)	Success Rate (%)	Duration(s) μ	Duration(s) σ
Route A	63.32	99	222.891	24.186
Route B	86.88	80	322.943	8.058
Route C	92.06	70	356.047	23.232

Indoor experiments

Two buildings, four routes :
Ten to fifteen dynamic obstacles are placed along the robot's route in building 2, while no dynamic obstacles is set in Building 1.



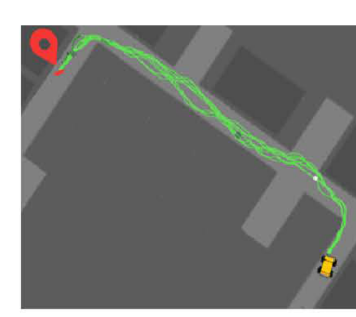
(a) Bldg 1 floorplan (b) Bldg 1 orb-slam (c) Bldg 2 floorplan (d) Bldg 2 orb-slam

Acheivements :
Achieves an average success rate of 95.3%.
Results show that our system possesses adaptability in different indoor environments that Husky can fit in.

Environment		Length (m)	Success Rate (%)	Duration(s) μ	Duration(s) σ
Building1	Route A	16.96	100	47.761	1.557
Building1	Route B	10.06	100	48.773	0.502
Building1	Route C	55.12	100	194.147	1.585
Building2	Route A	48.53	81.82	116.278	1.257

Other analysis

Analysis of the Trajectories Navigated by Husky
It is observed that the trajectories are similar for each case, qualitatively demonstrating the small variance in duration.
Performance in Range-Based Obstacle Avoidance Tasks
Average pedestrian density is up to 5.4 people per minute
This demonstrates that when our robot strays from the original path to avoid dynamic obstacles, virtual guidance is able to lead it back and resume its original task.



Building 2 (Route A)

Evaluation in obstacle avoidance tasks.

Route Length (m)	Success Rate (%)	Duration(s) μ	Duration(s) σ
10	100	36.77	1.569
20	83	74.09	3.144
40	80	114.40	3.284
100	77.8	397.196	8.652

Conclusion

We presented a straightforward, easy to implement, and effective modular framework using only a single monocular camera to handle the challenging robot navigation problem in the real world. We achieved this objective by introducing a virtual guidance scheme, which employs the virtual guide to navigate the robot's policy to its destination. We performed extensive experiments in diverse indoor and outdoor maps, and verified that our method is robust to various environmental conditions and generalizable to unfamiliar maps both in the virtual and real-world tasks.

More Info

Email us at cylee@cs.nthu.edu.tw

Watch Demo Video at QR code