Representations for Visual Navigation and How to Train Them

Saurabh Gupta
UIUC
In this talk, "Representations for Places that Afford Navigation in Novel Environments"
Basic Navigation Problems

Robot with a first person camera

Dropped into a novel environment

Discover paths or Explore

“Image Goal”

“Go Find a Chair”

“Explore the Environment”

Goal

“Go 300 feet North, 400 feet East”
Classical Solution

Hartley and Zisserman. 2000. Multiple View Geometry in Computer Vision

Kavraki et al. RA1996. Probabilistic roadmaps for path planning in high-
dimensional configuration spaces.
Lavalle and Kuffner. 2000. Rapidly-exploring random trees: Progress and
prospects.

Video Credits: Mur-Artal et al., Palmieri et al.
Geometric 3D Reconstruction of the World

Unnecessary

Do we need to tediously reconstruct everything on this table?

Humans can do quite a bit without accurate metric 3D information.

Perhaps, accurate full 3D is unnecessary?

Lacks Semantics

Speculating about space not directly observed.
Lacks Semantics

Eg: Finding a bathroom in a new restaurant
In this talk,

Representations for Places that Afford Navigation in Novel Environments

- Augmenting metric representations with semantic reasoning
- Relaxing the need for metric representations
- Scaling-up training of such representations

Operationalize insights from classical robotics into learning paradigms
Action to Execute

Spatial Representation of the World

Mapper

Goal (300, 400)

Spatial Representation of the World

Mapper

Planner

Neural Network

Action to Execute

Mapper

Planner

Goal (300, 400)

Mapper

Planner

Egomotion

Action to Execute
• Mapper and planner are differentiable functions
• Mapper and planner are learned for end task
• Hand-crafted obstacle maps to task-driven semantic maps
Spatial Representations

Feature $f_t$

Confidence $c_t$

Egocentric Bird's Eye Coordinate Frame
Differentiable Mapper

$\mathbf{f}_t$ $\mathbf{C}_t$ $\mathbf{f}_{t+1}$ $\mathbf{C}_{t+1}$

Egomotion

Encoder Decoder

Differentiable Warping
Differentiable Planner

Differentiable Planner

Differentiable Planner

Local computation that can be done using Convolutions and Channel-wise Max-Pooling.

Differentiable Planner

Local neighborhood tells about optimal action

Policy Training

Simulator based on scans of Real World Environments

Simulate robot views and motion

Compute ground truth traversability

Armeni et al. CVPR 2016. 3D Semantic Parsing of Large-Scale Indoor Spaces
Policy Training by Expert Imitation

Optimal Action

Predicted Action

Train with back-propagation
Results (Novel Env., Go To Relative Offset)

**Depth Input**

- Reactive
- LSTM
- Classical
- Our

**RGB Input**

- Reactive
- LSTM
- Classical
- Our
Goal (chair, table, door)
Semantic Tasks (Go to a chair)

Successful Navigations
by CMP
(Semantic Task)
Results (Novel Env, Go To Object)

**Depth Input**

<table>
<thead>
<tr>
<th>Method</th>
<th>Success Rate (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactive</td>
<td>20</td>
</tr>
<tr>
<td>LSTM</td>
<td>30</td>
</tr>
<tr>
<td>Classical</td>
<td>50</td>
</tr>
<tr>
<td>Our</td>
<td>80</td>
</tr>
</tbody>
</table>

**RGB Input**

<table>
<thead>
<tr>
<th>Method</th>
<th>Success Rate (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactive</td>
<td>10</td>
</tr>
<tr>
<td>LSTM</td>
<td>30</td>
</tr>
<tr>
<td>Classical</td>
<td>40</td>
</tr>
<tr>
<td>Our</td>
<td>70</td>
</tr>
</tbody>
</table>
Agent can make predictions about its surroundings
Agent can make predictions about its surroundings.

Free Space
Agent can make predictions about its surroundings
Representation for Places

- Spatial reasoning
- Semantic reasoning
  - Sensitive to pose error
  - Interactive training (DAgger easier than RL, but still)
  - Long training horizons

*Can we relax the need for spatially consistent global maps?*
Abstract

This paper studies the problem of image-goal navigation which involves navigating to the location indicated by a goal image in a novel previously unseen environment. To tackle this problem, we design topological representations for space that effectively leverage semantics and afford approximate geometric reasoning. At the heart of our representations are nodes with associated semantic features, that are interconnected using coarse geometric information. We describe supervised learning-based algorithms that can build, maintain and use such representations under noisy actuation. Experimental study in visually and physically realistic simulation suggests that our method builds effective representations that capture structural regularities and efficiently solve long-horizon navigation problems. We observe a relative improvement of more than 50% over existing methods that study this task.

1. Introduction

Imagine you are in a new house as shown in Fig 1 and you are given the task of finding a target object as shown in Fig 1 (top). While there are multiple possible directions to move, most of us would choose the path number 2 to move. This is because we use strong structural priors – we realize the target is an oven which is more likely to be found in the kitchen which seems accessible via path number 2. Now let us suppose, once you reach the oven, your goal is to reach back to the living room which you saw initially. How would you navigate? The answer to this question lies in how we humans store maps (or layout) of the house we just traversed. One possible answer would be metric maps, in which case we would know exactly how many steps to take to reach the living room. But this is clearly not how we humans operate [16, 41]. Instead, most of us would first get out of the kitchen by moving to the hallway and then navigate to the living room which is visible from the hallway. It is clear from the above examples, there are two main components of a successful visual navigation algorithm: (a) ability to build spatial representations and store them; (b) ability to exploit structural priors. When it comes to spatial representations, the majority of papers in navigation insist on building metrically precise representations of free space. However, metric maps have two major shortcomings: first, metric maps do not scale well with environment size and amount of experience. But more importantly, actuation noise on real-robots makes it challenging to build consistent representations, and precise localization may not always be possible. When it comes to exploiting structural priors, most learning-based approaches do not model these explicitly. Instead, they hope the learned policy function has these priors encoded implicitly. But it still remains unclear if these policy functions can encode semantic priors when learned via RL.

In this paper, we propose to tackle both the problems head-on. Instead of using metric-maps which are brittle to localization and noise, we propose a topological representation of the space. Our proposed representation consists of nodes that are connected in the form of a graph, based on local geometry information. Each node is represented by semantic features, and the edges encode coarse geometric information.

Reaching Image Goals

Image Goal Task

- Agent observations are panoramic images
- Take actions to navigate to the goal location
- Take the ‘stop’ action at the goal location
- Actuation noise, robot does not precisely know how much it has moved
**Geometric Prediction:**
Free space in different directions

**Semantic Prediction:**
Closeness to target

**Ghost Nodes**
0.8

**Relative Pose**

**Localization**

**Panorama**

**Current Observation**

**Goal Image**
4 Functions

- $F_G(I_1) = \text{Geometric Prediction: Free directions}$
- $F_S(I_1, I_2) = \text{Semantic Prediction: Closeness to target}$
- $F_R(I_1, I_2) = \text{Relative Pose}$
- $F_L(I_1, I_2) = \text{Localization}$
Using the Representation

Hierarchical Policy

High-Level Policy
- Decides where to go next and emits short-term goal
- Builds a topological map that keeps track of visited nodes, ghost nodes and values predicted by $F_s(I_1, I_2)$ for different directions

Low-Level Policy
- Executes actions to achieve short-term goal
- Option 1: predict local occupancy, plan paths
- Option 2: learn a low-level controller

Semantic Predictions via $F_s(I_1, I_2)$. 

Current Node

Past Nodes

Short-term Goal

$\Delta\text{Pose}$

Occupancy Map

FMM Cost Map

Forward
Left
Right
Stop
Building the Representation

Graph Update \((f_{GU})\)

Image Obs \((I_t)\)

Graph Localization \((\mathcal{F}_L)\)

Not localized

Node Addition

Localized

Agent’s Current Node
Regular Nodes
Ghost Nodes

Ghost Node addition

Graph \((G_t)\)
Building the Representation

Graph Update $f_{GU}$

Graph Localization $F_L$

Source Image $I_S$

Goal Image $I_G$

ResNet18 Encoder

Connection

$X_L$

Ghost Node addition

Graph Localization $F_L$

Node Addition

Image Obs $I_i$

Graph $G_{i-1}$

Agent's Current Node
Regular Nodes
Ghost Nodes

Not localized
Localized

$F_G$

Graph $G_i$
Building the Representation

Graph Update ($f_{GU}$)

Graph Localization ($\mathcal{F}_L$)

Graph ($G_{t-1}$)

Image Obs ($I_t$)

Agent’s Current Node
- Regular Nodes
- Ghost Nodes

Not localized

Ghost Node addition

Localzed

Graph ($G_t$)
Building the Representation

Graph Update \((f_{GU})\)

Geometric Explorable Area Prediction \((\mathcal{F}_G)\)

0 0 1 1 0 1 1 0 1 1 0 0
Single Supervised Learning Model

- \( F_G(I_1) = \text{Geometric Prediction: Free directions} \)
- \( F_S(I_1, I_2) = \text{Semantic Prediction: Closeness to target} \)
- \( F_R(I_1, I_2) = \text{Relative Pose} \)
- \( F_L(I_1, I_2) = \text{Localization} \)

- No reinforcement learning, no interaction needed
- Can be trained completely with static data
Goal Image

Start Location
Goal Location
Current Location
Regular Nodes
Ghost Nodes
Selected Ghost Node

Current Observation

$t = 20$
Goal Image

Start Location
Goal Location
Current Location
Regular Nodes
Ghost Nodes
Selected Ghost Node

$t = 27$

Current Observation
<table>
<thead>
<tr>
<th>Method</th>
<th>RGB</th>
<th>RGBD</th>
<th>RGBD (No Noise)</th>
<th>RGBD (No Stop)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM + Imitation</td>
<td>0.10</td>
<td>0.14</td>
<td>0.15</td>
<td>0.18</td>
</tr>
<tr>
<td>LSTM + RL</td>
<td>0.10</td>
<td>0.13</td>
<td>0.14</td>
<td>0.17</td>
</tr>
<tr>
<td>Occupancy Maps + FBE + RL</td>
<td></td>
<td></td>
<td>0.26</td>
<td>0.31</td>
</tr>
<tr>
<td>ANS</td>
<td>0.23</td>
<td>0.29</td>
<td>0.35</td>
<td>0.39</td>
</tr>
<tr>
<td>NTS (Our)</td>
<td>0.38</td>
<td>0.43</td>
<td>0.45</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Vanilla LSTM Memory

Metric Maps

Topological Maps

Map based methods are better than vanilla learning methods even in presence of noise.

NTS is better than occupancy map models, captures and uses semantic priors.

Robustness to Noise
In this subsection, we evaluate the proposed model on sequential goals in a single episode and study the importance of the topological map or the graph and the Semantic Score Predictor ($\mathcal{F}_S$). For creating a test episode with sequential goals, we randomly sample a goal between $1.5m$ to $5m$ away from the last goal. The agent gets a time budget of 500 timesteps for each goal. We consider two ablations: NTS w/o Graph. We pick the direction with the highest score in the current image greedily, not updating or using the graph over time. Intuitively, the performance of this ablation should deteriorate as the number of sequential goals increases as it has no memory of past observations. Neural Topological SLAM w/o Score Function. In this ablation, we do not use the Semantic Score Predictor ($\mathcal{F}_S$) and pick a ghost node randomly as the long-term goal when the Goal Image is not localized in the current graph. Intuitively, the performance of this ablation should improve with the increase in the number of sequential goals, as random exploration would build the graph over time and increase the likelihood of the Goal Image being localized.

We report the success rate and SPL of NTS and the two ablations as a function of the number of sequential goals in Figure 8. Success, in this case, is defined as the ratio of goals reached by the agent across a test set of 1000 episodes. Firstly, the performance of NTS is considerably higher than both the ablations, indicating the importance of both the components. The performance of all the models decreases with an increase in the number of sequential goals because if the agent fails to reach an intermediate goal, there is a high chance that the subsequent goals are farther away. However, the performance gap between NTS and NTS w/o Score Function decreases and the performance gap between NTS and NTS w/o Graph increases with increase in the number of sequential goals as expected. This indicates that the topological map becomes more important over time as the agent explores a new environment, and while the Semantic Score Predictor is the most important at the beginning to explore efficiently.
Representation for Places

- Spatial reasoning
- Semantic reasoning
  - Sensitive to pose error
  - Interactive training
  - Long training horizons
- Robust to pose error
- Offline supervised training
- Modularized policy

Can we simplify and scale-up training further?

$F_G(I_1)$: Geometry prediction
$F_R(I_1, I_2)$: Relative Pose
$F_L(I_1, I_2)$: Localization
$F_S(I_1, I_2)$: Semantic Prediction
Learning $F_s$ by Watching YouTube Videos

Basic Intuition

Mine for spatial co-occurrences

Video

Time

e.g. cues for finding a couch
Challenges in Using Such Videos

• Videos don’t come with action labels
  ➞ Action Grounding via an Inverse Model [1]

• Goals and intents are not known
  ➞ Use off-the-shelf object detectors to label frames with desired objects

• Depicted trajectories may not be optimal
  ➞ Use Q-learning to learn optimal behavior from sub-optimal data [2]


Value Learning from Videos (VLV)

a) Action Grounding
- Inverse Model: built by executing random actions on robot
- \( \hat{a}_{t-1} \) and \( \hat{a}_t \)

b) Goal Labeling
- Object Detector: trained on COCO
- Value function that uses implicitly learns semantic cues for seeking objects in novel indoor environments

\[ f(I, c) = \max_a Q^*(I, a, c) \]

Real Estate Tour from YouTube

Q-Learning Quadruple
- \( I_t \)
- \( I_{t+1} \)
- \( r_{t+1}^c \)
Learned Value Function

\[ f(I, c) \approx \text{nearness to goal} \]

Value function predicts a proxy for nearness to a goal object for a given image
Learned Value Function

\[ f(I, c) \approx \text{nearness to goal} \]

Value function predicts a proxy for nearness to a goal object for a given image
Using Learned Values for Semantic Navigation

Hierarchical Policy

High-Level Policy

• Decides where to go next and emits short-term goal
• Builds a topological map \([1]\) that stores values predicted by \(f(I, c)\) at different locations in different directions

Low-Level Policy

• Executes actions to achieve short-term goal
• Incrementally builds occupancy map from depth camera, plans paths

Results

[0.30] Topological Exploration
[0.46] Detection Seeker
[0.24] Behavior Cloning (YouTube)
[0.53] Ours (YouTube)
[0.28] RL
[0.23] BC (YouTube) + RL
Ablations

<table>
<thead>
<tr>
<th>Method</th>
<th>Easy</th>
<th>Medium</th>
<th>Hard</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Setting</td>
<td>0.62</td>
<td>0.42</td>
<td>0.23</td>
<td>0.40</td>
</tr>
<tr>
<td>True Actions</td>
<td>0.61</td>
<td>0.45</td>
<td>0.25</td>
<td>0.41</td>
</tr>
<tr>
<td>True Detections</td>
<td>0.62</td>
<td>0.45</td>
<td>0.22</td>
<td>0.40</td>
</tr>
<tr>
<td>True Rewards</td>
<td>0.64</td>
<td>0.46</td>
<td>0.21</td>
<td>0.41</td>
</tr>
<tr>
<td>Optimal Trajectories</td>
<td>0.65</td>
<td>0.46</td>
<td>0.25</td>
<td>0.43</td>
</tr>
<tr>
<td>Detector Score</td>
<td>0.73</td>
<td>0.48</td>
<td>0.26</td>
<td>0.46</td>
</tr>
<tr>
<td>Train on 360° Videos</td>
<td>0.66</td>
<td>0.51</td>
<td>0.32</td>
<td>0.47</td>
</tr>
<tr>
<td>No Hierarchy</td>
<td>0.38</td>
<td>0.10</td>
<td>0.02</td>
<td>0.15</td>
</tr>
</tbody>
</table>

- Inverse model and detector do not hurt performance significantly
- Detector at test time helps for close objects, panorama helps for far objects
- Q-Learning outperforms simple policy evaluation for challenging environments
- Hierarchical policy is a major factor in strong performance
Navigation to couches in novel environments
Representation for Places

- Spatial reasoning
- Semantic reasoning
- Robust to pose error
- Modularized policy
- Training from in-the-wild videos
In this talk,

**Representations for Places that Afford Navigation in Novel Environments**

- Augmenting metric representations with semantic reasoning
- Relaxing the need for metric representations
- Scaling-up training of such representations

**Operationalize insights from classical robotics into learning paradigms**