Review:

A Deep Learning Based Behavioral Approach to Indoor Autonomous Navigation

Patricio Cerda

About the paper

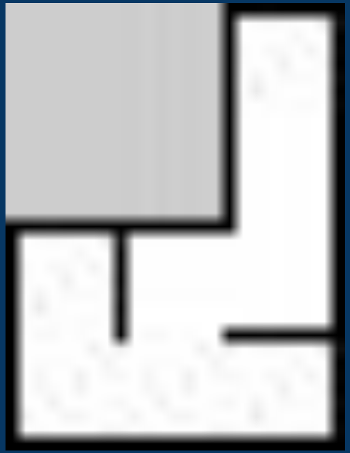
- Authors: G. Sepúlveda, J.C. Niebles, A. Soto
- Presented at ICRA 2018
- Great example of what we want for IA Nav group

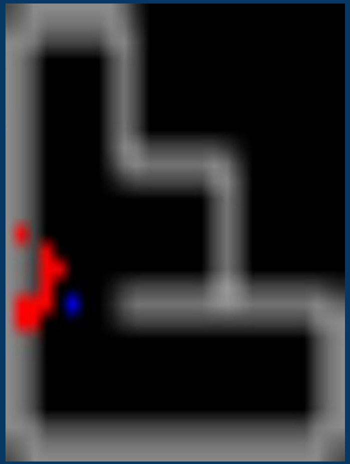
Robot Localization and Navigation

- Traditional approach? Subsumption, metric maps

- Example: see IIC2685 course

- Problem: non-optimal robustness
  - Odometry
  - Changes in geometry



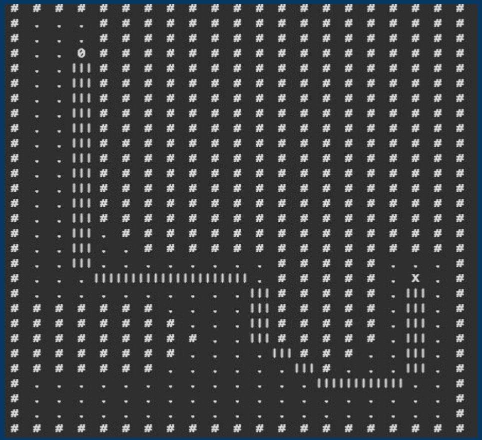


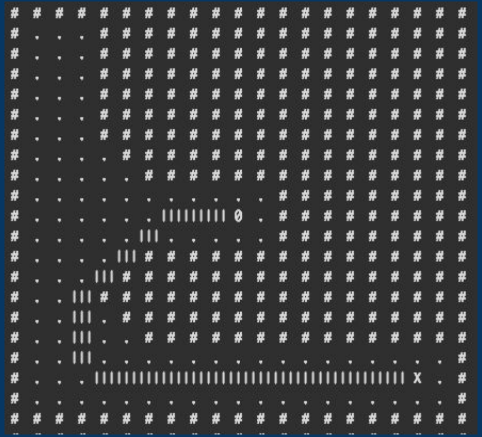
Robot Localization and Navigation

- Fundamentally different approach from humans

- Can we do better?

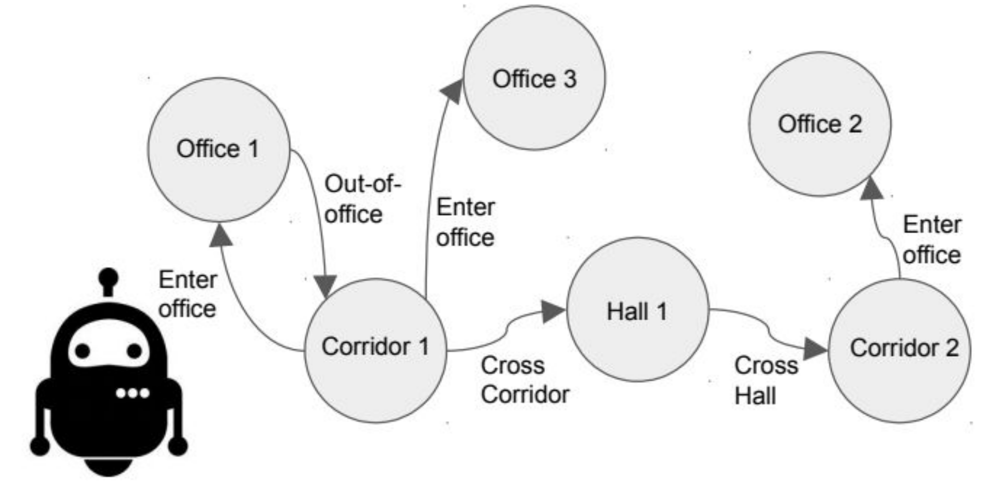
- Idea: leverage rich semantic structures from man-made environments





Graph-based map representation

- Nodes are places, identified by *perceptual* behaviors
- Edges are ‘basic’ *navigational* behaviors



Perception?

- Initial exploration phase to build a graph
- Deep Learning architecture to identify places (CNN)
- ...which then activate navigational behaviors

Benefits

- Robustness to localization errors: less dependance on local geometry

- Explicit internal world representation is deeply connected to execution of goal oriented behaviors

- Facilitates HRI, now the robot operates over a highly semantic world representation

Now, for the details...

Map representation

- Graph as a set of triplets: < place | behavior | place >

- In certain cases, the direction of approach implies distinction for two otherwise identical nodes

- Easy to use existing planning techniques over the graph

Navigational behaviors

- Robot should be able to robustly:
  - Leave an office
  - Enter an office
  - Follow a corridor
  - Cross a hall
  - ...among others

- Key to method's viability. Deep Learning to the rescue!

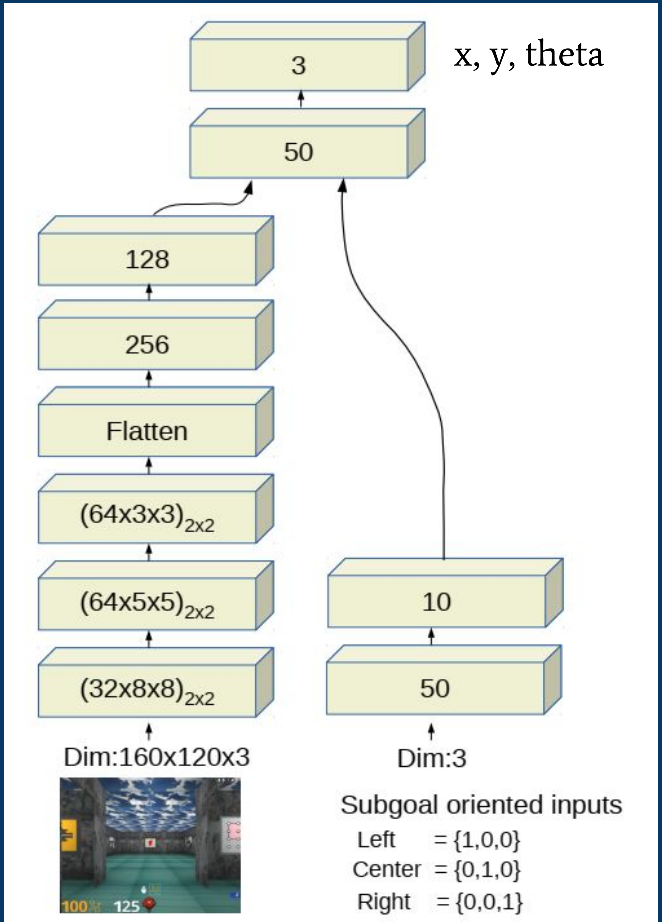
Navigational behaviors

- Supervised imitation learning based on deep CNNs
- Virtual environment: DeepMind's Lab, 3D, first person
- Traditional path-planning over the 2D map, agent then executes and records both actions and images.

Reactive behaviors

- They don't use robot's internal world knowledge => highly general

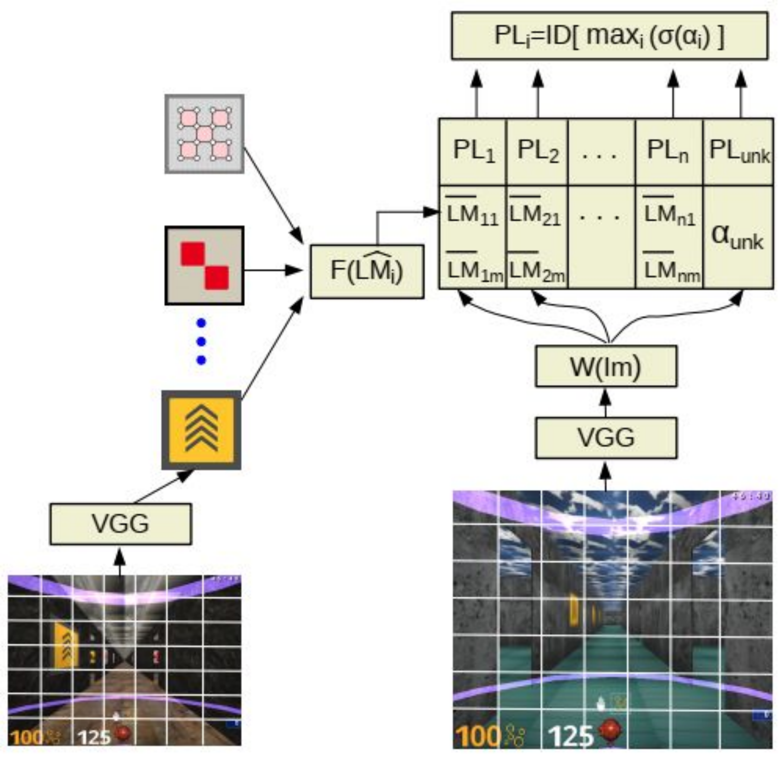
- Not purely reactive, paired up subgoal allows variations



CNN architecture

Memory-based behaviors

- They use robot's explicit internal representation
- Encapsulate specific knowledge from environment
- Place recognition => landmark detection, for simplicity



Architecture

Landmark detection

- For each unique landmark, get descriptors from pre-trained VGG-Net
- Train bilinear function to get embeddings, stored as keys, where the value is the place's ID
- Special ID to handle cases with no match ('unknown')

Landmark detection

- Cosine distance measures similarity between processed robot's view and available memory keys

$$\alpha_i = \sum_{l=1}^{7x7} \sum_{j=1}^{m} < Im_l, LM_{i,j} >$$

- Probability of being at each possible place: $p(PL_i) = \sigma(\alpha_i)$

$$\sigma(\alpha_i) = \frac{e^{\alpha_i}}{\sum_i e^{\alpha_i} + e^{\alpha_{unk}}}$$

Threshold for valid detection

- At testing, softmax is replaced by max()

Experiments

- Simulated offices present 3 possible structures: offices, corridors, halls

- Behaviors: manually defined. Could explore automatic techniques in future work!

Code	Description
ool	out of office, take left
oor	out of office, take right
cf	follow-corridor
iol	enter office to the left
ior	enter office to the right
chs	cross hall, continue straight
chl	cross hall, take left
chr	cross hall, take right
ccc	change corridor, straight
ccl	change corridor to the left
ccr	change corridor to the right

Training

- Grayscale images: faster than RGB, performs suitably well
- Adam optimizer. Random batches of 256 samples. LR=10e-4. Batch normalization for all but the last conv layer

Navigational Behaviors				
	Train: #paths	Train: #images	Test: #trials	Accuracy
cf	150	46131	100	100%
io	8000	221734	100	100%
oo	8000	259937	100	67%
oo-b	8000	259937	100	96%
ch	3000	248950	100	100%
cc	4000	123653	100	100%
Perceptual Behaviors				
	-	Train: #images	Test: #images	Class. Acc.
pd	-	12945	1500 (Imgs)	98.2%
lmpd	-	38459	1500 (Imgs)	96.7%

Detailed datasets + test accuracy

Training

- 100 generated maps. For each, 10 navigational tasks. The graph is directly fed (instead of discovering it). Good overall results validate approach!

#Maps	#Missions/map	#Missions	Av. Steps/mission	Acc.
100	10	1000	8.3	81.2%

- Main failures, both solvable in real implementation:
  - Occasional lack of office entrance detection, camera setup issue. Maybe backtracking variant could work?
  - Repeated texture patterns in virtual environment

Conclusions

- Proposed method integrates perceptual and navigational behaviors
  - Less prone to localization errors due to geometry changes, for example
  - Leverages semantically rich and compact world representation

- Pending challenge: implementation in real robots!