Deep Learning for Robot Perception and Navigation

Lifeng Bo, Claas Bollen, Thomas Brox, Andreas Eitel, Dieter Fox, Gabriel L. Oliveira, Oier Mees, Luciano Spinello, Jost Tobias Springenberg, Martin Riedmiller, Michael Ruhnke, Abhinav Valada, Jingwei Zhang, ...
Some facts about the AIS Lab

- 3 senior researchers
- 32 Ph.D. students
- >350 publications
- 1 technician
- 1 secretary
- 1 project manager
- Head of the DFG Cluster of Excellence BrainLinks-BrainTools
- Hosted in the Integrated Robotics Center
First Building Added by Robots to OpenStreetMap
The IRC Mapped by Robots
Freiburg is a Great Place

- Great team
- Great advisor
- Great city
- Great environment
- Great food
- Great wine
- Great 29 days of vacations
- Great 10 national holidays
- Great massage places

In case of any doubt: apply!
Fields of Research

- Mobile robotics
- State estimation and modeling
- Mapping
- Decision-theoretic approaches
- Adaptive techniques and learning
- Scene understanding
- Mobile manipulation
- Multi-robot coordination
- Robots and embedded systems
- Autonomous cars
- Flying vehicles
- ...
- Probabilistic robotics
Autonomous Robots
Robots that reliably fulfill their tasks in real-world environments
Neurobots
The Tagesthemen-Report
What our Robots Should do ...

- Perception
  - object recognition
  - human detection
  - sensor interpretation

- Navigation

- Manipulation
Why Deep Learning?

- Multiple layers of abstraction provide an advantage for solving complex (pattern recognition) problems
- Highly successful in computer vision and pattern recognition
- Can serve wide range of fields and applications
- End-to-end systems
Deep Learning in Robotics

- Robot perception is a challenging problem and involves many different aspects such as
  - Scene understanding
  - Object detection
  - Detection of humans

- Opportunities
  - improving perception,
  - manipulation
  - navigation
Multimodal Deep Learning for Robust RGB-D Object Recognition

Andreas Eitel, Jost Tobias Springenberg, Martin Riedmiller, Wolfram Burgard

[IROS 2015]
RGB-D Object Recognition
Often too little Data for Deep Learning Solutions

Deep networks are hard to train and require large amounts of data

- Lack of sufficient amount of labeled training data for RGB-D domain
- How to deal with limited sizes of available datasets?
Data often too Clean for Deep Learning Solutions

Large portion of RGB-D data is recorded under controlled settings

- How to improve recognition in real-world scenes when the training data is “clean”?
- How to deal with sensor noise from RGB-D sensors?
Solution: Transfer Deep RGB Features to Depth Domain

Both domains share similar features such as edges, corners, curves, ...
Solution: Transfer Deep RGB Features to Depth Domain

- Depth domain
- Pre-trained RGB CNN
- RGB domain

Transfer* to Depth Domain

- Depth encoding
- Fine-tune
- Re-train network features for depth

* Similar to [Schwarz et. al 2015, Gupta et. al 2014]
Solution: Transfer Deep RGB Features to Depth Domain

* Similar to [Schwarz et. al 2015, Gupta et. al 2014]
Multimodal Deep Convolutional Neural Network

- Two input modalities
- Late fusion network
- 10 convolutional layers
- Max pooling layers
- 4 fully connected layers
- Softmax classifier

2xAlexNet + fusion net
How to Encode Depth Images?

- Distribute depth over color channels
  - Compute min and max value of depth map
  - Shift depth map to min/max range
  - Normalize depth values to lie between 0 and 255
  - Colorize image using jet colormap (red = near, blue = far)

- Depth encoding improves recognition accuracy by 1.8 percentage points
Solution: Noise-aware Depth Feature Learning

"Clean" training data → Noise samples → Noise adaptation → Classify

Noise samples

"Clean" training data

Noise adaptation

Classify
Training with Noise Samples

Noise samples: 50,000

- Randomly sample noise for each training batch
- Shuffle noise samples

Input image
RGB Network Training

- Maximum likelihood learning
- Fine-tune from pre-trained AlexNet weights

\[ p(y \mid x) \]
Depth Network Training

- Maximum likelihood learning
- Fine-tune from pre-trained AlexNet weights

\[ p(y \mid d) \]
Fusion Network Training

- Fusion layers automatically learn to combine feature responses of the two network streams
- During training, weights in first layers stay fixed
UW RGB-D Object Dataset

Category-Level Recognition [%] (51 categories)

<table>
<thead>
<tr>
<th>Method</th>
<th>RGB</th>
<th>Depth</th>
<th>RGB-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-RNN</td>
<td>80.8</td>
<td>78.9</td>
<td>86.8</td>
</tr>
<tr>
<td>HMP</td>
<td>82.4</td>
<td>81.2</td>
<td>87.5</td>
</tr>
<tr>
<td>CaRFs</td>
<td>N/A</td>
<td>N/A</td>
<td>88.1</td>
</tr>
<tr>
<td>CNN Features</td>
<td>83.1</td>
<td>N/A</td>
<td>89.4</td>
</tr>
</tbody>
</table>

[Lai et al, 2011]
UW RGB-D Object Dataset

Category-Level Recognition [%] (51 categories)

<table>
<thead>
<tr>
<th>Method</th>
<th>RGB</th>
<th>Depth</th>
<th>RGB-D</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-RNN</td>
<td>80.8</td>
<td>78.9</td>
<td>86.8</td>
</tr>
<tr>
<td>HMP</td>
<td>82.4</td>
<td>81.2</td>
<td>87.5</td>
</tr>
<tr>
<td>CaRFs</td>
<td>N/A</td>
<td>N/A</td>
<td>88.1</td>
</tr>
<tr>
<td>CNN Features</td>
<td>83.1</td>
<td>N/A</td>
<td>89.4</td>
</tr>
<tr>
<td>This work, Fus-CNN</td>
<td><strong>84.1</strong></td>
<td><strong>83.8</strong></td>
<td><strong>91.3</strong></td>
</tr>
</tbody>
</table>

Confusion Matrix

<table>
<thead>
<tr>
<th>Label</th>
<th>Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>mushroom</td>
<td>garlic</td>
</tr>
<tr>
<td>pitcher</td>
<td>coffee mug</td>
</tr>
<tr>
<td>peach</td>
<td>garlic</td>
</tr>
</tbody>
</table>
Recognition in Noisy RGB-D Scenes

Recognition using annotated bounding boxes

Noise adapt. = correct prediction
No adapt. = false prediction

Category-Level Recognition [%] depth modality (6 categories)

<table>
<thead>
<tr>
<th>Noise adapt.</th>
<th>flash-light</th>
<th>cap</th>
<th>bowl</th>
<th>soda can</th>
<th>cereal box</th>
<th>coffee mug</th>
<th>class avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>97.5</td>
<td>68.5</td>
<td>66.5</td>
<td>66.6</td>
<td>96.2</td>
<td>79.1</td>
<td>79.1</td>
</tr>
<tr>
<td>√</td>
<td>96.4</td>
<td><strong>77.5</strong></td>
<td><strong>69.8</strong></td>
<td><strong>71.8</strong></td>
<td><strong>97.6</strong></td>
<td><strong>79.8</strong></td>
<td><strong>82.1</strong></td>
</tr>
</tbody>
</table>
Deep Learning for RGB-D Object Recognition

- Novel RGB-D object recognition for robotics
- Two-stream CNN with late fusion architecture
- Depth image transfer and noise augmentation training strategy
- State of the art on UW RGB-D Object dataset for category recognition: 91.3%
- Recognition accuracy of 82.1% on the RGB-D Scenes dataset
Choosing Smartly: Adaptive Multimodal Fusion for Object Detection in Changing Environments

Oier Mees,
Andreas Eitel, Wolfram Burgard
Object Detection in Changing Environments

- How to combine different modalities for detection
- Sensor noise changes with the environment

![RGB and Depth Images](image)
Mixture of Deep Neural Networks for People Detection

General learning approach to fuse different modalities such as RGB + depth + optical flow
Quantitative Results

- Comparison of fusion approaches
  - Late fusion approach with additional two-layer fusion network on top of expert networks
  - Hierarchical mixture of experts
  - Our Mixture of Experts approach (MoDE)
- Adaptive fusion improves performance

<table>
<thead>
<tr>
<th>Input</th>
<th>Method</th>
<th>AP/Recall</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth</td>
<td>HOD [Spinello et al. 2012]</td>
<td>-</td>
<td>56.3</td>
</tr>
<tr>
<td>RGB-D</td>
<td>HGE [Spinello et al. 2012]</td>
<td>-</td>
<td>87.4</td>
</tr>
<tr>
<td>RGB-D-Flow</td>
<td>Ours, CifarNet late fusion</td>
<td>88.0/88.4</td>
<td>88.2</td>
</tr>
<tr>
<td>RGB-D-Flow</td>
<td>Ours, MoDE</td>
<td><strong>88.6/90.0</strong></td>
<td><strong>89.3</strong></td>
</tr>
</tbody>
</table>
Adaptive Weighting in Test Set

Mean gating assignments per frame
Example Application
Qualitative Results

Gating assignments per bounding box
## Quantitative Results

### Performance of single and multimodal networks

<table>
<thead>
<tr>
<th>Input</th>
<th>Method</th>
<th>AP/Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>GoogLeNet-xxs</td>
<td>70.0/79.6</td>
</tr>
<tr>
<td>RGB</td>
<td>CifarNet</td>
<td>55.3/62.9</td>
</tr>
<tr>
<td>Depth</td>
<td>GoogLeNet-xxs</td>
<td>71.6/78.9</td>
</tr>
<tr>
<td>RGB-D</td>
<td>GoogLeNet-xxs average</td>
<td>71.1/73.9</td>
</tr>
<tr>
<td>RGB-D</td>
<td>GoogLeNet-xxs late fusion</td>
<td>72.0/76.3</td>
</tr>
<tr>
<td>RGB-D</td>
<td>GoogLeNet-xxs MoDE</td>
<td>80.4/81.1</td>
</tr>
</tbody>
</table>
Deep Learning for Human Part Discovery in Images

Gabriel L. Oliveira, Abhinav Valada, Claas Bollen, Wolfram Burgard, Thomas Brox

[to be presented at ICRA 2016]
Deep Learning for Human Part Discovery in Images

- Human-robot interaction

- Robot rescue
Deep Learning for Human Part Discovery in Images

- Dense prediction can provide pixel classification of the image
- Human part segmentation is naturally challenging due to
  - Non-rigid aspects of the body
  - Occlusions

PASCAL Parts  MS COCO  Freiburg Sitting
Network Architecture

- Fully convolutional network
  - Contraction and expansion of network input
  - Up-convolution operation for expansion
- Pixel input, pixel output
Experiments

- Evaluation of approach on
  - Publicly available computer vision datasets
  - Real-world datasets with ground and aerial robots
- Comparison against state-of-the-art semantic segmentation approach: FCN proposed by Long et al. [1]

Data Augmentation

Due to the low number of images in the available datasets, augmentation is crucial

- Spatial augmentation (rotation + scaling)
- Color augmentation
## PASCAL Parts Dataset

- **PASCAL Parts, 4 classes, IOU**

<table>
<thead>
<tr>
<th>Method</th>
<th>Head</th>
<th>Torso</th>
<th>Arms</th>
<th>Legs</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN</td>
<td>70.74</td>
<td>60.62</td>
<td>48.44</td>
<td>50.38</td>
<td>57.35</td>
</tr>
<tr>
<td>Ours</td>
<td>75.08</td>
<td>64.81</td>
<td>55.61</td>
<td>56.72</td>
<td>63.03</td>
</tr>
<tr>
<td>Ours (Spatial)</td>
<td>80.49</td>
<td>74.39</td>
<td>67.17</td>
<td>70.39</td>
<td>73.00</td>
</tr>
<tr>
<td>Ours (Spatial + Color)</td>
<td><strong>83.24</strong></td>
<td><strong>79.41</strong></td>
<td><strong>73.73</strong></td>
<td><strong>76.52</strong></td>
<td><strong>78.23</strong></td>
</tr>
</tbody>
</table>

- **PASCAL Parts, 14 classes, IOU**

<table>
<thead>
<tr>
<th>Method</th>
<th>Head</th>
<th>Torso</th>
<th>L U arm</th>
<th>L LW arm</th>
<th>L hand</th>
<th>R U hand</th>
<th>R LW arm</th>
<th>R hand</th>
<th>R LW leg</th>
<th>R foot</th>
<th>L U leg</th>
<th>L LW leg</th>
<th>L foot</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN</td>
<td>74.0</td>
<td>66.2</td>
<td>56.6</td>
<td>46.0</td>
<td>34.1</td>
<td>58.9</td>
<td>44.1</td>
<td>31.0</td>
<td>49.3</td>
<td>44.5</td>
<td>40.8</td>
<td>48.5</td>
<td>47.6</td>
<td>41.2</td>
</tr>
<tr>
<td>Ours (Spatial)</td>
<td>81.8</td>
<td>78.0</td>
<td>69.5</td>
<td>63.1</td>
<td>59.0</td>
<td>71.2</td>
<td>63.0</td>
<td>58.7</td>
<td>65.4</td>
<td>60.6</td>
<td>52.0</td>
<td>67.9</td>
<td>60.3</td>
<td>50.0</td>
</tr>
<tr>
<td>Ours (Spatial+Color)</td>
<td><strong>84.0</strong></td>
<td><strong>81.5</strong></td>
<td><strong>74.1</strong></td>
<td><strong>68.0</strong></td>
<td><strong>64.0</strong></td>
<td><strong>75.4</strong></td>
<td><strong>67.4</strong></td>
<td><strong>61.9</strong></td>
<td><strong>72.4</strong></td>
<td><strong>67.1</strong></td>
<td><strong>56.9</strong></td>
<td><strong>73.0</strong></td>
<td><strong>66.1</strong></td>
<td><strong>57.7</strong></td>
</tr>
</tbody>
</table>

\(R = \text{right}, \ L = \text{left}, \ U = \text{upper}, \ LW = \text{lower}\).
Freiburg Sitting People Part Segmentation Dataset

- We present a novel dataset for human part segmentation in wheelchairs.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN</td>
<td>59.69</td>
<td>43.17</td>
</tr>
<tr>
<td>Ours (Trained on PASCAL)</td>
<td>78.04</td>
<td>59.84</td>
</tr>
<tr>
<td>Ours (2 people train - 4 people test)</td>
<td>81.78</td>
<td>64.10</td>
</tr>
</tbody>
</table>
Robot Experiments

- Range experiments with ground robot
- Aerial platform for disaster scenario (Segmentation under severe body occlusions)
Range Experiments

Recorded using Bumblebee camera

- Robust to radial distortion
- Robust to scale

(a) 1.0 meter
(b) 2.0 meters
(c) 3.0 meters
(d) 4.0 meters
(e) 5.0 meters
(f) 6.0 meters

![Graph showing Mean IOU (%) vs Distance (m)]
Freiburg People in Disaster

Dataset designed to test severe occlusions

<table>
<thead>
<tr>
<th>Method</th>
<th>Head</th>
<th>Torso</th>
<th>Arms</th>
<th>Legs</th>
<th>IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN</td>
<td>52.71</td>
<td>62.49</td>
<td>35.04</td>
<td>43.25</td>
<td>43.20</td>
</tr>
<tr>
<td>Ours</td>
<td>80.56</td>
<td>79.45</td>
<td>63.93</td>
<td>64.91</td>
<td>71.99</td>
</tr>
</tbody>
</table>
Application to Obelix Data
Efficient Deep Models for Monocular Road Segmentation

Gabriel Leivas Oliveira

Wolfram Burgard
Thomas Brox
University of Freiburg, Germany
Motivation
Architecture

More parameters expansion
1-to-C*Ncl filters per refinement
Terrain Classification using a Late Fusion DCNN Architecture

- Snow
- Glare
- Low Lighting
Autonomous Navigation in Outdoor Areas
Terrain Classification using a Late Fusion DCNN Architecture
Semantic Segmentation of Moving Objects using Convolutional Neural Networks

Johan Vertens, Abhinav Valada, Wolfram Burgard
Goal

1. Robust and fast semantic segmentation of driving scenarios
2. Semantic motion segmentation

For semantic motion segmentation we consider the “car”-class.
Semantic Motion Segmentation

- Fuse semantic features and generate motion features within a CNN
- Two architectures:
  1. FiltFlow-Net: Takes precomputed motion features
  2. Siamese-Net: Motion features are learned entirely
FiltFlow-Net

Consecutive Images

Optical Flow (DeepFlow)

Motion GT

Depth

Predicted Flow

Filtered Flow
FiltFlow-Net: Architecture

- Embedded MultiNet
- Predicts moving cars
Siamese-Net: Architecture
Comparison of Architectures

- FiltFlow-Net achieves an improvement of 6.26 IoU over Siamese-Net

<table>
<thead>
<tr>
<th>Approach</th>
<th>IoU</th>
<th>AP</th>
<th>FPR</th>
<th>FNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>FiltFlow-Net</td>
<td>83.44</td>
<td>94.67</td>
<td>04.39</td>
<td>11.41</td>
</tr>
<tr>
<td>Siamese-Net</td>
<td>77.18</td>
<td>89.64</td>
<td>09.10</td>
<td>13.68</td>
</tr>
</tbody>
</table>
## Comparison of Inference Time

- Siamese-Net has much lower inference time

<table>
<thead>
<tr>
<th></th>
<th>FiltFlow-Net</th>
<th>Siamese-Net</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optical Flow (DeepFlow)</td>
<td>11.2s</td>
<td>-</td>
</tr>
<tr>
<td>Predicted Optical Flow</td>
<td>112ms</td>
<td>-</td>
</tr>
<tr>
<td>Neural Network</td>
<td>87.4ms</td>
<td>83.3ms</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>11.4s</strong></td>
<td><strong>83.3ms</strong></td>
</tr>
</tbody>
</table>
KITTI Motion Segmentation

FiltFlow-Net

Driving Car  Standing Car  Car behind max-range
Cityscapes Motion Segmentation

FiltFlow-Net

Driving Car  Standing Car  Car behind max-range
Deep Feature Learning for Acoustics-based Terrain Classification

Abhinav Valada, Luciano Spinello, Wolfram Bugard
Motivation

Optical sensors are highly sensitive to visual changes

- Lighting Variations
- Shadows
- Dirt on Lens
Motivation

Use sound from vehicle-terrain interactions to classify terrain
Network Architecture

- Novel architecture designed for unstructured sound data
- Global pooling gathers statistics of learned features across time
Data Collection

- Wood
- Linoleum
- Carpet
- Asphalt
- Mowed Grass
- Grass
- Paving
- Cobble Stone
- Offroad

P3-DX
## Results - Baseline Comparison

(300ms window)

<table>
<thead>
<tr>
<th>Features</th>
<th>SVM Linear</th>
<th>SVM RBF</th>
<th>k-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ginna [1]</td>
<td>44.87 ± 0.70</td>
<td>37.51 ± 0.74</td>
<td>57.26 ± 0.60</td>
</tr>
<tr>
<td>Spectral [2]</td>
<td>84.48 ± 0.36</td>
<td>78.65 ± 0.45</td>
<td>76.02 ± 0.43</td>
</tr>
<tr>
<td>Ginna &amp; Shape [3]</td>
<td>85.50 ± 0.34</td>
<td>80.37 ± 0.55</td>
<td>78.17 ± 0.37</td>
</tr>
<tr>
<td>MFCC &amp; Chroma [4]</td>
<td>88.95 ± 0.21</td>
<td><strong>88.55 ± 0.20</strong></td>
<td>88.43 ± 0.15</td>
</tr>
<tr>
<td>Trimbral [5]</td>
<td>89.07 ± 0.12</td>
<td>86.74 ± 0.25</td>
<td>84.82 ± 0.54</td>
</tr>
<tr>
<td>Cepstral [6]</td>
<td><strong>89.93 ± 0.21</strong></td>
<td>78.93 ± 0.62</td>
<td><strong>88.63 ± 0.06</strong></td>
</tr>
</tbody>
</table>

---

**18.92% improvement over window previous state of the art**

Real-World Stress Testing

Avg. accuracy of **98.54%**

- True Positives
- False Positives
Can You Guess the Gerrain?

Social Experiment

- Avg. human performance = 24.66%
- Avg. network performance = 99.5%
- Go to deepterrain.cs.uni-freiburg.de
- Listen to five sound clips of a robot traversing on different terrains
- Guess what terrain they are
Liquid Height Detection in Cups using RGB-D Data is Hard

**Transparent Liquid:** Water  
**Opaque Liquid:** Orange juice

![Images showing refracted bottom and reflected liquid height]
Approach Overview

- Extract edge and view angle features
- Extract liquid level and view angle features
- Apply refraction conversion
- Probabilistic RGB model
- Probabilistic Depth model
- Joint probabilistic model
Average Fluid Level Error

- Our Joint Model
- Hara 2014
- Naïve
Opaque vs. Transparent Classification

Correctly Classified (%) vs. Fill level (%)

- 25%
- 50%
- 75%
- 100%
Pouring

**Liquid level detection assumes:**
- Liquid is present in the cup
- Can take multiple views of the liquid at the same height

**Pouring challenges:**
- Starting with empty cup so that no initial liquid to obtain information from
- Only single view of cup so that it is awkward to take multiple views while pouring

**Assumption:**
- Liquid type is known
Liquid level Estimation during Pouring

- **Opaque Liquid**: Use raw measured depth value

- **Transparent Liquid**: Liquid height can be determined from

\[
h_r = \left( 1 - \frac{\cos(\alpha)}{\sqrt{n_i^2 - 1 + \cos^2(\alpha)}} \right) h
\]
Tracking the Liquid Level using a Kalman Filter

- Depth measurement - blue
- Kalman filtered height - red

Opaque Liquid - Infrared light is reflected
Tracking the Liquid Level using a Kalman Filter
... and End to End Navigation
Deep Reinforcement Learning with Successor Features for Navigation across Similar Environments
Motivation

- Finding a solution for navigation that:
  - Does not require explicit SLAM, localization and path planning procedures
  - Can adapt to new situations (new navigation goals and environments)

- Aim for the agent:
  - Capable of solving all tasks by the end of training
  - Using minimal interaction time for each task
Transfer learning between navigation tasks

Transfer learning scenarios:

1. Multiple goal positions: same environment and transition dynamics but different reward function.
2. Multiple environments: changes in the maze structure or robot dynamics
Successor Feature RL with Task Transfer (SF-RL-Transfer)

\[
\phi^k_{s_t} = \phi^k(s_t, \theta^k_{\phi})
\]

\[
Q^\pi_k(s, a) \approx \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \phi^k_{s_t} \mid s_0 = s, a_0 = a, \pi^k \right] \cdot \omega^k
\]

\[
\phi^i_s = \mathcal{B}^i \phi^k_s, \mathcal{B}^k = I, \forall i \leq k
\]

\[
Q^\pi_i(s, a) \approx \mathbb{E} \left[ \sum_{t=0}^{\infty} \mathcal{B}^i \gamma^t \phi^k_{s_t} \mid s_0 = s, a_0 = a, \pi^i \right] \cdot \omega^i
\]

\[
= \mathcal{B}^i \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t \phi^k_{s_t} \mid s_0 = s, a_0 = a, \pi^i \right] \omega^i
\]

\[
= \mathcal{B}^i \psi^\pi_i (\phi^k_{s_t}, a)^T \omega^i
\]

\[
= \psi^\pi_i (\mathcal{B}^i \phi^k_{s_t}, a)^T \omega^i.
\]
Training Setup

<table>
<thead>
<tr>
<th>Synthetic Depth Images</th>
<th>Real Depth Images</th>
</tr>
</thead>
</table>

- Synthetic Depth Images
- Real Depth Images
SF-RL-Transfer
Real-world Experiments

Map5

3D Model of a Maze-like World
Overall Conclusions

- Deep networks are a promising approach to solve complex perception problems in robotics
- The key challenges are
  - finding the proper architecture
  - using proper data augmentation strategies
- Goal: Achieving end-to-end learning of complex (navigation) tasks.
What is the Future of Probabilistic Robotics?
Thank you for your attention!