MapNet: An allocentric spatial memory for mapping environments

João F. Henriques, Andrea Vedaldi
Visual Geometry Group
Motivation

What we usually have:

- Object detections
- Segmentations
- 3D information (relative to camera)
- ...

⇒ Image-centric tasks
Motivation

What we would like:

• Reason beyond image, into world
• Object permanence
• Eventually, long-term goals and planning

⇒ World-centric tasks
Simultaneous Localization And Mapping (SLAM)

Time

Frame #1
Agent
Map
Location

Frame #2
Agent
Map
Location

Frame #3
Agent
Map
Location

Classic SLAM
(No learning)

• Hard to adapt to new environments (hand-tuning)
• No semantic information
• No use of priors to compensate for missing data
Related work – deep learning for SLAM

- No map
- Cannot correct for inevitable drift

Egomotion predictors:

Costante’15, Clark’17, Zhu’17, Wang’17, ...
Related work – deep learning for SLAM

- MapNet, CVPR 2018

Offline-learned localization

- Map is stored in deep network's parameters
- New environments require re-training

Kendall'15, Mirowski'18, Brahmbhatt'18, ...
Related work – deep learning for SLAM

- Map is created on-the-fly as activations
- Perfect egomotion input is used for localization, not map
- Tested on synthetic environments (so far)

*Online mapping, no localization*

*Kanitscheider’16, Gupta’17, Zhang’17, Parisotto’17, ...*
Proposed method

- Performs **both Mapping and Localization** with a deep net
- No egomotion information
- Fully online (mapping as we go)

Henriques and Vedaldi, *MapNet*, CVPR 2018
Allocentric map memory

Map model:
- Represent **ground plane** as 2D grid.
- Store one **embedding** per location.
- Allows associating semantics with world coordinates.
Localization and mapping as dual operators

Core insight: Localization $\Leftrightarrow$ convolution  
Mapping $\Leftrightarrow$ deconvolution
Ground projected CNN features

- Given depth and camera intrinsics, **project** CNN features to ground-plane.
- Since camera pose is unknown, the output 2D grid is **local** (camera-space).

Henriques and Vedaldi, *MapNet*, CVPR 2018
Localization

Localize by **dense matching** of the local view's embeddings to the map.

- Requires only **one** cross-correlation (convolution).
- Can be interpreted as addressing a **spatial associative memory**.
Also consider **camera orientation**:

- Simply resample the local view at several rotations.
- Use as **filter bank** for cross-correlation.
Localization

Local view → Resampler (rotation) → Rotated local views → Cross-correlation ← Softmax → Position and orientation heatmap → Orientations

Camera reference-frame → Map → World reference-frame

Henriques and Vedaldi, *MapNet*, CVPR 2018
The **mapping** step updates the map with the local view.

- The local view must be **registered** to world-space.
- Requires one **deconvolution** of the position/orientation heatmap, using the local views (filter bank).

- After registration, the local view can be easily integrated into the map (e.g. by linear interpolation, or a convolutional LSTM)
Henriques and Vedaldi, MapNet, CVPR 2018
Henriques and Vedaldi, MapNet, CVPR 2018

Full pipeline

Image → CNN → Ground projection → Local view → Resampler (rotation) → Mapping ↔ deconvolution

Localization ↔ convolution

Map → Localization → Position and orientation heatmap → Registered local view

→ LSTM → Updated map
Experiments – 2D data

Toy problem setup

- 100,000 mazes
- Agent moves at random
- Limited, local visibility

Training

- Input sequences of 5 frames
- Position/orientation supervision
- Min. logistic loss of predicted position (heatmap)
Experiments – 2D data

Global view

Local view (always facing right)

Predicted heatmap (blue – ground truth)
Experiments – 2D data

**Global view**

**Local view** (always facing right)

**Predicted heatmap** (blue – ground truth)
Experiments – 2D data

Map tensor (one channel per column)

Sample #1

Sample #2

Sample #3

Sample #4

⇒ Several local views are integrated into a larger map.
Experiments – 2D data

Is this map semantic? → Yes!

- Assigned class labels to maze cells (corridors, turns, dead-ends...).
- Class label is correctly predicted from a cell’s embedding most of the time.

<table>
<thead>
<tr>
<th>Corridor</th>
<th>Turn</th>
<th>Dead end</th>
<th>Fork</th>
<th>Crossroad</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>76.1%</td>
<td>73.3%</td>
<td>69.8%</td>
<td>68.8%</td>
<td>62.3%</td>
<td>71.3%</td>
</tr>
</tbody>
</table>

Balanced dataset prediction accuracy (chance: 50%)
Experiments – 3D game data

ResearchDoom Dataset

- 4 recorded speed-runs through the whole game
- 6 hours of gameplay
- Challenging, large hand-crafted levels

https://www.youtube.com/watch?v=mInSO7YW1EU
Experiments – 3D real data

Active Vision Dataset

- Robot platform in 19 indoor scenes
- Images collected at all positions/orientations
- Can be composed into unlimited sequences

https://www.youtube.com/watch?v=-MUXfcrxGEM
Experiments – 3D data quantitative results

**ResearchDoom Dataset**

- MapNet-32
- MapNet-32-F
- CNN-LSTM-128
- CNN-LSTM-1024
- DeepVO-128
- DeepVO-1024

**Active Vision Dataset**

- MapNet-32
- MapNet-32-F
- CNN-LSTM-128
- CNN-LSTM-1024
- DeepVO-128
- DeepVO-1024
- ORB-SLAM2
Conclusions

- We perform SLAM entirely online using an end-to-end learned architecture.
- **Localization** and **Mapping** are a dual pair of convolution/deconvolution.
- **Semantic** embeddings of the World arise from the self-localization objective.
- **Next step**: navigation and long-term goals.

Project page with code: www.robots.ox.ac.uk/~joao/mapnet