Learning-based Visual Localization

Alex Kendall, University of Cambridge

CVPR 2017 tutorial on Large-Scale Visual Place Recognition and Image-Based Localization
Alex Kendall, Torstel Sattler, Giorgos Tolias, Akihiko Torii
What is the motivation to use machine learning to localise?
Are point-based feature descriptors the right landmarks?
Also, should maps be Euclidean?

• Not all places are of equal importance?
• Can we learn a better ‘map’?
Biological learning for localisation

2014 Nobel Prize in Physiology or Medicine for the discovery of place and grid cells
[O'Keefe, Edvard and May-Britt Moser]

- Located in the hippocampus
- **Place cells** encode topological and hierarchical location
- **Grid cells** encode Euclidean space for precise positioning and path finding
Machine Learning for Localisation

Feature based localisation (previous talk)

Scene Coordinate Regression (Part 1)

Pose Regression (Part 2)

Feature Extraction

Landmark Registration

6DOF Pose Estimation

Machine learning
Part 1: Scene Coordinate Regression
Part 1: Scene Coordinate Regression

Scene coordinate XYZ ↔ RGB color space
Scene Coordinate Localisation

• Infer scene coordinates for each pixel location in a test image

Scene Coordinate Regression

• Use SLAM/ Kinect Fusion / etc to generate ground truth labels
• Train a regression forest to regress scene coordinates
• Train on depth-aware features using RGB-D data

Jamie Shotton et al. Scene coordinate regression forests for camera relocalization in RGB-D images. CVPR 2013
Generate Camera Pose Hypothesis

Jamie Shotton et al. Scene coordinate regression forests for camera relocalization in RGB-D images. CVPR 2013
RANSAC to infer camera pose

Jamie Shotton et al. Scene coordinate regression forests for camera relocalization in RGB-D images. CVPR 2013
Datasets – Seven Scenes – Indoor Localization

• 17,000 images across 7 small indoor scenes.

Jamie Shotton et al. Scene coordinate regression forests for camera relocalization in RGB-D images. CVPR 2013
Outperforms SIFT-Feature methods

**Metric:**
Proportion of test frames with < 0.05m translational error and < 5° angular error

<table>
<thead>
<tr>
<th>Scene</th>
<th>Baselines</th>
<th></th>
<th></th>
<th></th>
<th>Our Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tiny-image RGB-D</td>
<td>Sparse RGB</td>
<td>Depth</td>
<td>DA-RGB</td>
<td>DA-RGB + D</td>
</tr>
<tr>
<td>Chess</td>
<td>0.0%</td>
<td>70.7%</td>
<td>82.7%</td>
<td><strong>92.6%</strong></td>
<td>91.5%</td>
</tr>
<tr>
<td>Fire</td>
<td>0.5%</td>
<td>49.9%</td>
<td>44.7%</td>
<td><strong>82.9%</strong></td>
<td>74.7%</td>
</tr>
<tr>
<td>Heads</td>
<td>0.0%</td>
<td><strong>67.6%</strong></td>
<td>27.0%</td>
<td>49.4%</td>
<td>46.8%</td>
</tr>
<tr>
<td>Office</td>
<td>0.0%</td>
<td>36.6%</td>
<td>65.5%</td>
<td>74.9%</td>
<td><strong>79.1%</strong></td>
</tr>
<tr>
<td>Pumpkin</td>
<td>0.0%</td>
<td>21.3%</td>
<td>58.6%</td>
<td>73.7%</td>
<td>72.7%</td>
</tr>
<tr>
<td>RedKitchen</td>
<td>0.0%</td>
<td>29.8%</td>
<td>61.3%</td>
<td>71.8%</td>
<td><strong>72.9%</strong></td>
</tr>
<tr>
<td>Stairs</td>
<td>0.0%</td>
<td>9.2%</td>
<td>12.2%</td>
<td><strong>27.8%</strong></td>
<td>24.4%</td>
</tr>
</tbody>
</table>

SIFT feature registration
Different features for regression forest
Further Literature


- Neural networks improve scene coordinate regression but not for RANSAC optimization


- Scene coordinate regression with RGB only images


- End-to-end learning through RANSAC with deep learning

---

Input Image | Ground Truth Scene Coordinates | Predicted Scene Coordinates
Score Regression Conclusions

✓ Map no longer Euclidean but learned features
✓ Scales efficiently with scene size
✗ Features don’t use global context
✗ Inliers are noisy and RANSAC not always reliable
Part 2: Pose Regression with Deep Learning
Trained with a naïve end-to-end loss function to regress camera position, $x$, and orientation, $q$

$$\text{loss}(I) = \|x - \hat{x}\|_2 + \beta \left\| q - \frac{\hat{q}}{\|\hat{q}\|} \right\|_2$$

Camera Pose Regression

training data in green, test data in blue, PoseNet results in red

Tolerance to environment, unknown intrinsics, weather, etc.

Robust in scenarios where SIFT-Feature localisation fails


SIFT based registration fails, but deep learning features are able to localise!

<table>
<thead>
<tr>
<th>Area</th>
<th># train/test</th>
<th>PoseNet [23]</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>5575 m²</td>
<td>875/220</td>
<td>1.87 m, 6.14°</td>
<td>1.31 m, 2.79° (30,55)</td>
</tr>
</tbody>
</table>
Does our network recognise context?

- Saliency maps show significance of each pixel w.r.t. localisation
- PoseNet learns to ignore dynamic objects and recognises large context and contours around landmarks

PoseNet’s performance improves with more data

Scales very well:
- Constant inference time (single forward pass of the network)
- Constant memory (~5 MB of neural network weights)

PoseNet Summary

✓ Robust to lighting, weather, dynamic objects
✓ Fast inference, <2ms per image on Titan GPU
✓ Scale not dependent on number of training images

✗ Coarse accuracy
✗ Difficult to learn both position vs orientation

Intolerant to weighting between position and orientation regression loss

\[
\text{loss}(I) = \|x - \hat{x}\|_2 + \beta \left\| q - \frac{\hat{q}}{\|\hat{q}\|} \right\|_2
\]

Learning camera pose, \textit{with geometry}

Train with reprojection loss of 3-D geometry using predicted and ground truth camera poses.

\[
loss(I) = \frac{1}{|G'|} \sum_{g_i \in G'} \|\pi(q, x, g_i) - \pi(\hat{q}, \hat{x}, g_i)\|_\gamma
\]

Where \( \pi \) is the projection function of 3-D point \( g_i \)

Automatically learns a weighting between position, \( x \), and orientation, \( q \).

Datasets – Cambridge Landmarks – Outdoor Localization

- 8,000 images from 6 scenes up to 100 x 500m

Datasets – Seven Scenes – Indoor Localization

• 17,000 images across 7 small indoor scenes.

Jamie Shotton et al. Scene coordinate regression forests for camera relocalization in RGB-D images. CVPR 2013
Datasets – Dubrovnik – Large Scale Localization

• 6000 images across 1500 x 1500 m in Dubrovnik, Croatia.
• Varying weather, season, camera type

## Geometry Improves Performance

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Median Error</th>
<th>Accuracy</th>
<th>Median Error</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear sum, $\beta = 500$ [1]</td>
<td>1.52 m, 1.19 $^\circ$</td>
<td>65%</td>
<td>13.1 m, 4.68 $^\circ$</td>
<td>30.1%</td>
</tr>
<tr>
<td>Learn weighting with task uncertainty [2]</td>
<td>0.99 m, 1.06 $^\circ$</td>
<td>85.3%</td>
<td>9.88 m, 4.73 $^\circ$</td>
<td>41.7%</td>
</tr>
<tr>
<td>Reprojection loss [2]</td>
<td></td>
<td><strong>does not converge</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learn weighting pretrain + Reprojection loss [2]</td>
<td>0.88 m, 1.04 $^\circ$</td>
<td>90.3%</td>
<td>7.90 m, 4.40 $^\circ$</td>
<td>48.6%</td>
</tr>
<tr>
<td>SIFT + SfM Geometry [3]</td>
<td>0.42 m, 0.55 $^\circ$</td>
<td>-</td>
<td>1.1 m, -</td>
<td>-</td>
</tr>
</tbody>
</table>


Future Work & What’s Next?

- PoseNet is much faster and requires smaller images than traditional methods

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>King’s College</td>
<td>0.88m, 1.04°</td>
<td>0.42m, 0.55°</td>
</tr>
<tr>
<td>Resolution</td>
<td>256 x 256 px</td>
<td>1920 × 1080 px</td>
</tr>
<tr>
<td>Inference Time</td>
<td>2 ms</td>
<td>78 ms</td>
</tr>
</tbody>
</table>

- Can we scale model towards city scale localisation with deep learning?
- How to improve fine grained accuracy for accurate registration?

Further Improvements to Camera Pose Regression

- Modelling uncertainty: Kendall et al. ICRA 2017
- Video localisation: Clark et al. CVPR 2017
- Ego-motion estimation: Melekhov et al. arXiv 2017
- RGB-D localisation: Li et al. IEEE Transactions 2017
Increase feature context with spatial LSTMs

Improves metric localisation performance from PoseNet by 5-50% (depending on the dataset)

Modeling Aleatoric Uncertainty with Probabilistic Deep Learning

Use probabilistic modelling to learn data-dependant uncertainty with no additional supervision.

<table>
<thead>
<tr>
<th></th>
<th>Deep Learning</th>
<th>Probabilistic Deep Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td>$\hat{y} = f(x)$</td>
<td>$[\hat{y}, \sigma^2] = f(x)$</td>
</tr>
<tr>
<td><strong>Regression</strong></td>
<td>$\text{Loss} = |y - \hat{y}|^2$</td>
<td>$\text{Loss} = \frac{|y - \hat{y}|^2}{2\sigma^2} + \log \sigma^2$</td>
</tr>
</tbody>
</table>

Model Uncertainty with Dropout

• Use dropout to learn a distribution over models
• Sample using Monte Carlo dropout sampling a test time to obtain posterior distribution

Uncertainty to estimate loop closure

We can use epistemic uncertainty to estimate metric relocalisation error.

Determine if the model has seen the landmark before (loop closure).

Increased uncertainty from strong occlusion, motion blur, visually ambiguous landmarks.

Video Localisation

- PoseNet + Temporal Recurrent Neural Network
  - Learns dynamics of platform - temporal features
  - Bidirectional - analogous to “smoothing”
- Mixture of Gaussian output

Video Localisation Improves Temporal Consistency

- Outperforms smoothing baseline
- Diminishing returns using very long sequences

Camera Pose Regression with RGB-D Sensors

<table>
<thead>
<tr>
<th>7 Scenes Data Results</th>
<th>Median Position</th>
<th>Median Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoseNet RGB</td>
<td>0.52m</td>
<td>12.8</td>
</tr>
<tr>
<td>Dual Stream PoseNet + Depth</td>
<td>0.35m</td>
<td>10.2</td>
</tr>
</tbody>
</table>

Conclusions + Further Research Questions?

✓ Learning can produce **more efficient map** representations

✓ Convolutional nets are **more robust** to environmental challenges

? **Accuracy** still not as good as traditional feature methods

? How do we learn **city-scale localisation** map with deep learning?

? How can we **learn and update a map online**?

? How do we obtain **large city-scale datasets** with accurate pose labels?
Thank you! Questions?


alexgkendall.com  ❯ Slides available soon!

PoseNet Webdemo:
mi.eng.cam.ac.uk/projects/relocalisation/