3D Pose Regression using Convolutional Neural Networks

Siddharth Mahendran  Haider Ali  René Vidal
Center for Imaging Science, Johns Hopkins University

Motivation and Problem Statement
Motivation: 3D pose estimation is a key component of challenging vision problems like scene understanding and autonomous navigation

Problem statement: Given a 2D image and a bounding box around an object in the image, estimate the 3D rotation \( R \) between the object and the camera

Introduction and Related Work
Prior work discretizes the pose space into key poses and treats the pose estimation problem as a classification problem

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation</td>
<td>Discretized angles (21 bins)</td>
<td>Discretized angles (360 bins)</td>
<td>Axis-angle</td>
</tr>
<tr>
<td>Loss function</td>
<td>Cross-entropy</td>
<td>Weighted cross-entropy</td>
<td>Geodesic loss</td>
</tr>
<tr>
<td>Data augmentation</td>
<td>2D jittering</td>
<td>Rendered images</td>
<td>3D pose jittering</td>
</tr>
<tr>
<td>Network architecture</td>
<td>VGG-Net (FC7)</td>
<td>AlexNet (FC7)</td>
<td>VGG-M (FC6)</td>
</tr>
</tbody>
</table>

Contribution: Instead of breaking up pose space into discrete key poses, we propose a regression formulation using representations (Axis-angle and Quaternion), loss functions (Geodesic loss between rotation matrices) and data augmentation techniques (3D pose jittering) that respect and exploit the non-Euclidean structure of the space of rotations.

Representing 3D Rotations
Rotation matrices lie in the Special Orthogonal group:
\[
SO(3) = \{ R \in \mathbb{R}^{3 \times 3} : R^T R = I_3, \det(R) = 1 \} \quad (1)
\]

Geodesic loss function on the space of rotation matrices:
\[
\mathcal{L}(R, R_i) = \frac{|| \log R_i R_i^T ||}{\sqrt{2}} \quad (2)
\]

Geodesic loss between ground-truth and predicted rotations:
\[
\mathcal{L}(R, \hat{R}) = | \cos^{-1} \left( \frac{1}{2} \text{trace}(R \hat{R}^T) - 1 \right) | \quad (3)
\]

Geodesic loss between ground-truth and predicted quaternions:
\[
\mathcal{L}(q, \hat{q}) = \cos^{-1} \left( \langle \hat{q}, q \rangle \right) \quad (4)
\]

3D Pose Jittering
For every image, 3D pose annotations of azimuth \( \text{az} \), elevation \( \text{el} \) and camera-tilt \( \text{ct} \) give 3D rotation \( R(\text{az}, \text{el}, \text{ct}) \). We perturb around ground-truth 3D pose using these transformations:

- Flips:
  \( R(\text{-az}, \text{el}, \text{-ct}) \)
- In-plane rotations:
  \( R(\text{az}, \text{el}, \text{ct} \pm \delta) \)
- Out-of-plane rotations:
  \( R(\text{az} \pm 2\text{az}, \text{el}, \text{ct}) \)

Network Architecture

Ablation Analysis

Experiments
- Dataset: Pascal3D+ [2] consists of ImageNet and Pascal VOC2012 images with 3D pose annotations. ImageNet trainval, VOC2012 train, and VOC2012-val images are used as training, validation, and testing data respectively.
- Training: Two step learning procedure: (i) Train the pose networks (with feature network fixed) using augmented and rendered data, and (ii) Finetune the overall network using original and flipped images

Contributions
- Representing 3D Rotations
- Geodesic loss
- Axis-angle
- Quaternion
- Network architectures
- Ablation analysis
- Experiments

This work was supported by NSF grant 1527340.