3D Poses in the Wild (3DPW) Challenge
Winning Approach
ECCV 2020

Presented by István Sárándi (RWTH Aachen University)

Based on the methods from

Task: 3D Human Pose Tracking

• Given an RGB video, estimate the 3D pose of each annotated person in each frame

• Two variations:
  – **Known association**: Estimated poses can be matched to 2D GT in each frame
  – **Unknown association**: IDs can be matched only in the first frame (using 2D reference), to determine who to track
Our Approach

- **Detect** people per frame with an off-the-shelf detector (YOLOv3)
- **Estimate** absolute 3D pose for each detection (MeTRAb)
- **Associate** based on pose distance (naive frame-to-frame tracking)
  - Either to the predicted poses of the previous frame (in camera coords)
  - Or to reference 2D pose (if allowed)
Our 3D Pose Estimator

- **Volumetric heatmaps** are a powerful representation
  - Introduced in [Pavlakos et al. ‘17] with a coarse-to-fine estimation scheme
  - Combined in [Sun et al. ‘18] with soft-argmax heatmap decoding
  - Our contributions
    - Further simplification: directly predict low-res (8x8) heatmap with a standard backbone (e.g. ResNet), no need for any additional learned layers
    - Define the heatmap axes in a novel way...
Metric-Scale Truncation-Robust Heatmaps

- **Limitations** of the (2.5D) heatmap approach:
  - **Scale recovery**: The X and Y coordinates are in image (pixel) space, but we want metric-scale predictions in millimeters (common: non-learned post-processing step)
  - **Truncation**: No way to estimate body joints outside the image boundaries

- Direct numerical coordinate regression would not suffer from these problems
- **However**, heatmap prediction fits better to the convolutional structure and leads to more accurate localization

- **Question**: Can we have the best of both?
  - Recover a full metric-scale pose while staying in the heatmap paradigm?
Metric-Scale Truncation-Robust (MeTRo) Heatmaps

- Define the heatmap axes directly in the 3D metric space, irrespective of image zooming
Metric-Scale Truncation-Robust Heatmaps

• Benefit:
  – No need for heuristic scale recovery, the backbone is trained to estimate people’s size implicitly
  – We always get a complete 3D pose

• Downside:
  – No 2D image space pose, only 3D root-relative
  – As heatmap peaks are not predicted at their image space position, localization might be slightly less precise
Metric-Scale Truncation-Robust Heatmaps

- **How can this even work** if input and output are not aligned?
  - Receptive fields are large enough (signal can move to new position as we go deeper in the network)
  - Backbone learns a scaling transformation
  - Truncation is sensed by the network through the zero padding at the borders (c.f. [1])

MeTRAbs

• Now we can recover **3D metric-scale complete** poses, without having to give up the heatmap idea

• But we lost the alignment to the image space

• However, we can jointly estimate **both 2D and 3D** heatmaps:
  - negligible extra computational cost: $8*J \rightarrow 9*J$ channels
  - if the camera is calibrated, we can even recover the camera-space (**absolute**) 3D coordinates!
MeTRAbs Architecture
MeTRAbs

- Despite its simplicity, state-of-the-art results on
  - Human3.6M
  - MPI-INF-3DHP
  - MuPoTS-3D
- Fast execution:
  - ~500 crops per second (ResNet-50, 256x256 px, stride 32, batch size 8, 2080Ti)
Qualitative results

H36M

H36M (partial body)

3DHP
Qualitative Results (MuPoTS-3D)
Training Data for the Challenge

- Many 3D pose datasets released in recent years
- **Large-scale supervised learning** is thus possible
  - Human3.6M – 164,528 [number of sufficiently different poses (thresh. 100 mm)]
  - MuCo-3DHP – 676,875 [includes repeated poses in different composites]
  - CMU-Panoptic – 858,390
  - SURREAL – 1,577,006
  - SAILVOS – 90,611
- For better generalization, **weak supervision** from 2D datasets is also important
  - COCO, MPII, LSP, ...
Dataset Merging

- Learn jointly from all datasets with **mixed batches**:
  - 36 examples from the real 3D datasets (H36M, MuCo, CMU)
  - 12 examples from SAILVOS
  - 8 examples from SURREAL
  - 8 examples from COCO (2D only)
- All datasets use somewhat **different joint definitions**
- 3DPW benchmark requires SMPL body joints (24 keypoints)
- Goal: use **all available supervision** but keep final output **specific to SMPL**
  - Merge joints across datasets that are sufficiently similar (e.g. wrists)
  - Do not merge others (e.g. define multiple hip joints, one per dataset)
  - We use a total of 73 distinct joints in the model
  - Some SMPL joints are only supervised through SURREAL
Dataset Merging

- Redundant joints, but allows dataset-specific benchmarking
Strong Data Augmentation

- Crucial for generalization to in-the-wild scenes
  - Synthetic occlusions (paste object segments onto the image)
  - Background replacement (segment the training images beforehand if masks not given in dataset)
  - Color distortion
  - Geometric transformations (scale, flip, shift, rotate)
- Test-time augmentation: 5 crops
## Results

**Known association**

<table>
<thead>
<tr>
<th>#</th>
<th>User</th>
<th>Entries</th>
<th>Date of Last Entry</th>
<th>Team Name</th>
<th>Rank</th>
<th>MPJPE</th>
<th>MPIPE_PA</th>
<th>PCK</th>
<th>AUC</th>
<th>MPIAE</th>
<th>MPIAE_PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>isarandi</td>
<td>2</td>
<td>08/01/20</td>
<td></td>
<td>1.000</td>
<td>72.3786</td>
<td>53.0564</td>
<td>47.3431</td>
<td>0.6624</td>
<td>-14</td>
<td>-14</td>
</tr>
<tr>
<td>2</td>
<td>DJ_Walker</td>
<td>7</td>
<td>08/22/20</td>
<td>JDAI-CV</td>
<td>3.000</td>
<td>81.7641</td>
<td>58.6131</td>
<td>37.3293</td>
<td>0.5991</td>
<td>20.8089</td>
<td>19.0901</td>
</tr>
<tr>
<td>3</td>
<td>milo</td>
<td>12</td>
<td>08/01/20</td>
<td>milo</td>
<td>3.2500</td>
<td>83.1544</td>
<td>59.7027</td>
<td>42.1494</td>
<td>0.6231</td>
<td>19.6965</td>
<td>19.1486</td>
</tr>
<tr>
<td>4</td>
<td>rbr</td>
<td>12</td>
<td>08/20/20</td>
<td>SNU CVLAB</td>
<td>4.2500</td>
<td>83.1845</td>
<td>64.1717</td>
<td>46.9092</td>
<td>0.6323</td>
<td>20.1264</td>
<td>19.9578</td>
</tr>
<tr>
<td>5</td>
<td>mks0601</td>
<td>16</td>
<td>08/01/20</td>
<td>SNU CVLAB</td>
<td>6.0000</td>
<td>84.2889</td>
<td>61.7516</td>
<td>36.6064</td>
<td>0.5966</td>
<td>21.2543</td>
<td>19.7324</td>
</tr>
</tbody>
</table>

**Unknown association**

<table>
<thead>
<tr>
<th>#</th>
<th>User</th>
<th>Entries</th>
<th>Date of Last Entry</th>
<th>Team Name</th>
<th>Rank</th>
<th>MPJPE</th>
<th>MPIPE_PA</th>
<th>PCK</th>
<th>AUC</th>
<th>MPIAE</th>
<th>MPIAE_PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>isarandi</td>
<td>2</td>
<td>08/02/20</td>
<td></td>
<td>1.000</td>
<td>83.3594</td>
<td>58.8521</td>
<td>44.6241</td>
<td>0.6315</td>
<td>-9</td>
<td>-9</td>
</tr>
<tr>
<td>2</td>
<td>mks0601</td>
<td>12</td>
<td>08/01/20</td>
<td>SNU CVLAB</td>
<td>2.7500</td>
<td>86.3566</td>
<td>63.1444</td>
<td>36.2311</td>
<td>0.5908</td>
<td>22.2005</td>
<td>20.3347</td>
</tr>
<tr>
<td>3</td>
<td>root9527</td>
<td>4</td>
<td>08/20/20</td>
<td>SNU CVLAB</td>
<td>3.7500</td>
<td>87.0182</td>
<td>61.4503</td>
<td>36.0264</td>
<td>0.5888</td>
<td>22.2413</td>
<td>19.3888</td>
</tr>
<tr>
<td>4</td>
<td>redarknight</td>
<td>16</td>
<td>08/01/20</td>
<td>SNU CVLAB</td>
<td>3.7500</td>
<td>87.8796</td>
<td>64.2319</td>
<td>36.4411</td>
<td>0.5890</td>
<td>24.2251</td>
<td>21.3447</td>
</tr>
<tr>
<td>5</td>
<td>zenluo</td>
<td>2</td>
<td>08/02/20</td>
<td></td>
<td>5.5000</td>
<td>91.5124</td>
<td>67.6933</td>
<td>33.9416</td>
<td>0.5655</td>
<td>23.5770</td>
<td>20.6931</td>
</tr>
</tbody>
</table>
Results

• Robust predictions even under occlusion and bad illumination
Discussion

- Overall recipe:
  - Formulate the task such that a CNN can predict the desired output (here: metric-space complete pose)
  - In a well-suited representation (here: heatmap)
  - Then supervise it with strongly augmented, diverse examples from many sources by carefully merging datasets

- Limitations:
  - Single-frame estimator, no temporal smoothing
  - Needs a separate person detector
  - Naive pose matching, no ReID (still not many ID switches on 3DPW)
Thank you!

- Inference code and pretrained model available (self-contained model file → just a few lines of code to run!)
  - https://github.com/isarandi/metrabs
- Also thanks to my co-authors of the underlying papers!

István Sárándi  Timm Linder  Kai O. Arras  Bastian Leibe