Mehta, Sotnychenko, Mueller et al.

Single-Shot Multi-Person 3D Body Pose Estimation From Monocular RGB Input

1 Summary

Mehta et al. describe a method for multi-person 3D pose estimation which uses an occlusion-robust pose-map (ORPM) formulation. This formulation, which extends 3D location-maps originally designed for single-person tracking, enables full-body pose inference even in cases with occlusion and has a fixed number of outputs (therefore supporting multi-person estimation via part associations). The authors also introduce a dataset of sophisticated multi-person interactions, created by compositing single-person data.

2 Introduction

- This paper proposes a DNN-based method to estimate the 3D poses of multiple people from a single image. Notably, the images can involve occlusion.
  - the method jointly infers 2D and 3D locations (of all people in the scene) in a single forward pass
  - the method does not require explicit bounding box detection and is fully convolutional
  - pose representation: new multi-person 3D pose-maps (store full 3D pose of each person at the corresponding detected 2D torso location) which can model strong occlusions
- It also describes the synthesis of a multi-person 3D dataset **MuCo-3DHP** (containing ground truth 3D pose) via the composition of single-person images.
- Their method is state-of-the-art for multi-person, and performs competitively on single-person.

3 Multi-Person Dataset

- The authors create MuPoTS-3D, an in-the-wild multi-person 3D pose benchmark (test set).
  - "MuPoTS-3D" stands for **Mu**lti-Person **Po**se **T**est **S**et in **3D**
  - contains real, filmed, not composited data
- They also create MuCo-3DHP (a larger training set), which composites real single-person data into an arbitrary number of multi-person interaction images under user control (with 3D pose annotations).
  - "MuCo-3DHP" stands for **Mu**lti-Person **Co**mposited **3D** **H**uman **P**ose

4 Method

- **Overall, 3D pose estimation proceeds as follows:**
  1. Predict 2D joint locations, 2D part affinity fields, and 3D pose maps in a single shot.
  2. Use 2D joint locations and 2D part affinity fields to extract skeletons for all people in the image.
  3. Use 3D pose maps to *read-out* the 3D poses for each skeleton.
- The method builds on the location-map formulation, which uses image evidence to infer 3D pose by inferring 3D joint positions at respective 2D joint locations.

  - **location-map:**
    - three for every joint
    - contains the 3D x, y, or z coordinate of a joint at the joint's 2D pixel location
    - estimate these along with a 2D pixel location heatmap for each joint
  - **occlusion-robust pose-map:**
    - a new encoding of 3D pose for multiple people
    - incorporates redundancy into location-maps with a special read-out scheme
    - allows read-out of complete limb pose at any 2D limb joint, allows read-out of complete base pose at any torso joint (have pose be obtainable from the spots that end up visible)
- The algorithm jointly predicts n heatmaps, $3n$ pose-maps, and $2n$ part affinity fields.
  - n is the number of joints
- It then uses these for pose inference.
  - 2D pose estimation is akin to OpenPose.
  - 3D read-out scheme:
    - Start at neck, get full base pose from there. (If neck is not visible, try pelvis. If pelvis is not visible, treat as "not a person.")
    - Attempt to refine each limb pose by going to the limb and looking for a pose there. For each limb (head, arms, legs), go to the extremity joint (head, wrists, ankles), take the limb pose from it, and use it to override the corresponding current limb pose in the output skeleton.
      - If the extremity joint isn't visible, go up the kinematic tree (e.g. elbow, from wrist) and look for a limb pose there. Stop when we get one, or after we go through the whole limb.
    - *Motivation behind redundancy*: some parts might not be visible due to occlusion. Also, it's better to use a limb joint's estimation of the limb pose if possible, because it's more local and accurate (base poses given by the neck are usually close to the average in the dataset, i.e. not that good).
    - *Summary: go to neck. Get pose from there. Then go to extremities. If each extremity exists, use extremity pose for limb. If it doesn't exist, walk up the limb's kinematic tree and repeat.*

5 Results and Discussion

- It works well.

6 Clarifications

- *Problem with full redundancy* (i.e. full pose at each limb), besides efficiency: if two joints are close together, their heatmaps in the confidence map might be treated as one joint and we won't know which of the two people to encode (or read out) at that joint. Also, w.r.t. location-maps, there are many more inter-joint conflicts that can happen with full-body redundancy.

References

[1] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt. Single-Shot Multi-Person 3D Body Pose Estimation From Monocular RGB Input. *arXiv:1712.03453 (2017).*