Zhou, Huang, Sun, Xue, Wei

Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach

1 Summary

- *The goal*: get computers to estimate the 3D poses of humans in the wild.
- *Available resources*: in-the-wild dataset with 2D annotations, in-the-lab dataset with 3D annotations.
- *Solution*: train an end-to-end system with a 2D module and a 3D module, mix strong and weak 3D supervision along with a geometry-based regularization.
- *The result*: a network (output: 2D and 3D pose) that exploits the correlation between 2D pose estimation and depth estimation while also taking into account the image.

This is the architecture:

- Start with a batch of images from both 2D (in-the-wild) and 3D (not in-the-wild) datasets.
- Pass the images through some convolutional layers.
- Pass the convolutional features through a stacked hourglass 2D pose estimation module.
  - **(Get 2D pose.)**
- Pass the 2D joint heatmaps and intermediate 2D features through a depth regression module.
  - **(Get 3D pose.)**
  - By involving intermediate features instead of just the output of the 2D module, the 3D module obtains extra semantic information from the image that it can use. There are multiple 3D interpretations of a single 2D skeleton, so 2D pose isn't always enough; we want to use image cues too.

This is the loss:

- $\mathcal{L}(\hat{Y}_{hm}, \hat{Y}_{depth}|I) = \mathcal{L}_{2D}(\hat{Y}_{hm}, Y_{2D}) + \mathcal{L}_{depth}(\hat{Y}_{depth}|I, Y_{2D})$ where $\hat{Y}_{hm}$ is the predicted heatmaps, $\hat{Y}_{depth}$ is the predicted depth, I is the image, and Y_{2D} is the ground-truth 2D.
- $\mathcal{L}_{2D}(\hat{Y}_{hm}, Y_{2D})$ is the 2D loss, and is simply the L_2 distance between the predicted heatmaps and the heatmaps rendered from the ground-truth 2D through a Gaussian kernel.
- $\mathcal{L}_{depth}(\hat{Y}_{depth}|I, Y_{2D})$ is the depth regression loss, and is an L_2 distance between predicted and ground-truth if I comes from the 3D dataset, and a special **geometric loss** if I comes from the 2D dataset.
- The geometric loss is based on the idea that the bone ratios $l_e/\bar{l}_e$ in a person should remain constant, and therefore is defined as the sum of *variances of bone ratios within each of four skeleton groups.*
  - l_e is the length of bone e in 3D (for one person)
  - $\bar{l}_e$ is the average length of bone e over all 3D skeletons
  - want to minimize the variance of bone ratios $l_e/\bar{l}_e$

This is the training `tl;dr`:

- Send a bunch of images in.
- Predict 2D and 3D for all.
- Supervise with 2D loss on the 2D output, and also (for the 3D output)...

- If the image was originally from the 3D dataset, supervise with L_2 loss on real annotations.
- If the image was originally from the 2D dataset, weakly supervise with geometric 3D loss.

This is a miscellaneous detail:

- It's hard to train from scratch. The paper follows a three-stage training scheme – training first the 2D module, then the full network without geometric loss, and finally the full network with geometric loss.

This is a more holistic `tl;dr`:

- There's a 2D module, which estimates the pixel coordinates of each joint, connected to a depth regression module, which estimates the mm depth of each joint. The system is trained end-to-end on strong supervision when it's available, and otherwise on a weak geometric supervision which makes the bone ratios stay the same. The final network is able to predict the 2D and 3D poses of humans in the wild.

References

[1] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, Yichen Wei. Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach. *arXiv preprint arXiv:1704.02447 (2017).*

Appendix: Geometric Loss

The geometric loss for a single skeleton is

$$\sum_i var\left(\text{bone ratios in } R_i\right) = \sum_i \frac{1}{|R_i|} \sum_{e \in R_i} \left(\frac{l_e}{\bar{l}_e} - \bar{r}_i\right)^2$$

where R_i is a bone group (and i is therefore the index specifying a bone group), and

$$\bar{r}_i = \text{mean bone ratio in } R_i = \frac{1}{|R_i|} \sum_{e \in R_i} \frac{l_e}{\bar{l}_e}$$

Here, we minimize variance (average squared distance from the mean bone ratio), which is essentially saying "make sure that the lengths of this person's limbs with respect to the average are consistent." We don't want the lengths to *be* the average; we want them to consistently be some *scaling* of the average within a bone group.

For example, a person's forearm length should be the same with respect to the average forearm length as the person's upper arm length with respect to the average upper arm length. With the given groups

- R_{arm}: left forearm, left upper arm, right forearm, right upper arm
- R_{leg}: left lower leg, left upper leg, right lower leg, right upper leg
- $R_{shoulder}$: left shoulder bone, right shoulder bone
- R_{hip}: left hip bone, right hip bone

we would be optimizing the leg group (e.g.) to approach

$$\frac{l_{\text{left lower leg}}}{\bar{l}_{\text{left lower leg}}} = \frac{l_{\text{left upper leg}}}{\bar{l}_{\text{left upper leg}}} = \frac{l_{\text{right lower leg}}}{\bar{l}_{\text{right lower leg}}} = \frac{l_{\text{right upper leg}}}{\bar{l}_{\text{right upper leg}}}$$

If we had poses for **one person** across multiple frames, we could similarly optimize for the skeleton to have consistent bone lengths between frames by minimizing the L_2 distance between the bone length in each frame and the average length of that bone across all of the frames. This is a stronger objective than the paper's, and says "*make sure the lengths of the person's bones do not change from one frame to the next.*"

Appendix: One Last `tl;dr`

- Use stacked hourglass module to predict 2D heatmaps; supervise with L2 against ground truth heatmaps.
- Use depth regression module to predict z; supervise w/ ground truth z if available, else geometric loss.