It's all Relative: Monocular 3D Human Pose Estimation from Weakly Supervised Data - Supplementary Material

Matteo Ruggero Ronchi*, Oisin Mac Aodha*, Robert Eng, Pietro Perona
{mronchi,macaodha,reng,perona}@caltech.edu
California Institute of Technology

1 Implementation Details

We use the same fully connected network architecture as [2] for all experiments. We use the one stage version of the model (Fig. 1) as we only observed a minor loss in performance compared the two stage version. To predict the scale parameter s used in our reprojection loss we add an additional fully connected layer to the output of the penultimate set of layers and apply a sigmoid non-linearity to its output. The output of the non-linearity is scaled using a hyperparameter r to allow the network to predict an arbitrarily wide range. For the default method 'Ours Relative', detailed in Tab. 1 of the main paper, we set $r = 1$. In the relative depth loss we set $\lambda = 2.5$. We set the weighting hyperparameters α and γ in the main loss to 1.0 and set β to 0.1. We train all models on Human3.6M for 25 epochs, as we observe that they do not tend to benefit from additional training time. We train our relative model from scratch on LSP for 100 epochs. For our relative model we center the input 2D keypoints by setting the root location to (0,0). We did not perform this centering for the supervised baseline as we found that it hurt performance, but we did center the 3D coordinates in a similar fashion. As in [2], we clip the gradients to 1.0 during training. Training time on Human3.6M is less than five minutes for one epoch for our relative model.

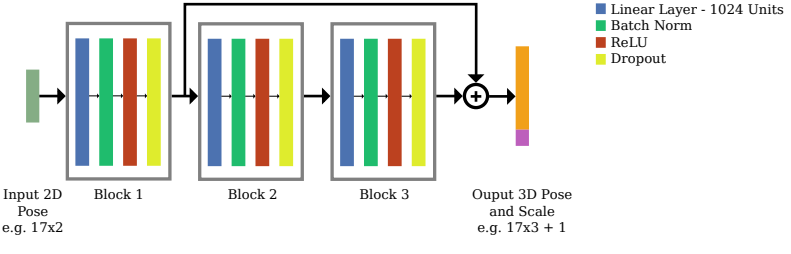
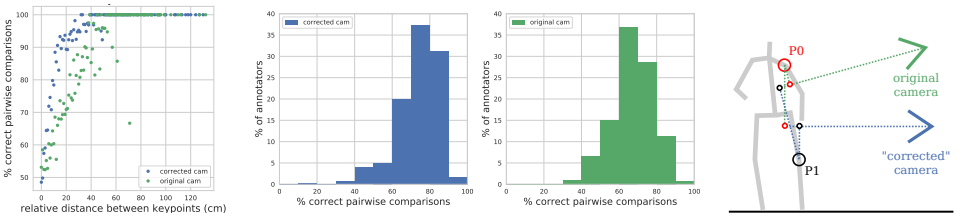


Figure 1: Network architecture. We use a similar architecture to [2] but include scale prediction at the end of the network.

*These authors contributed equally to this work.



(a) Accuracy vs Distance (b) Annotator Accuracy (c) Camera Correction

Figure 2: Human relative depth annotation performance on 1,000 images selected from the Human3.6M dataset [1]. (a-b) Without correcting for the orientation of the camera, annotators perform worse (green lines). (c) The green camera represents the input view and the blue is the upright orientated view as perceived by our annotators. If the camera is orientated upwards when performing the evaluation the relative depths are a better match to the annotator provided labels.

2 Human Annotation Performance

In this section we provide some additional discussion of the results of our user study on Human3.6M [1]. As mentioned in the main paper, we observed that annotators tended to estimate depth in images by correcting for the orientation of the camera. In Fig. 2 (c) we see an illustration of this effect. Here, from the perspective of the original camera view in green the keypoint 'P0' is closer than 'P1'. In practice, even though they see an image of the scene taken from the perspective of the green camera, annotators seemingly correct for the orientation of the camera and 'imagine' the distance of the scene from the perspective of the blue camera. While this change in camera position is subtle, it affects the relative ordering of the points as 'P1' is now closer to the camera. We hypothesize that this is a result of the annotator imagining themselves in the same pose as the individual in the image and then estimating the distance to the camera in a Manhattan world sense. Without correcting for this effect 67% of the provided pairwise annotations are correct, but when this is taken into account then accuracy increases to 71%. We correct for the bias by forcing the camera to be upright when computing the scene depth. The results before and after applying this correction and annotator accuracies can be viewed in Figs. 2 (a) and (b). This effect is likely to be exacerbated in Human3.6M as there are only four different camera viewpoints in the entire dataset and they are all facing downwards. We expect this to be less of an issue for datasets that feature a larger variation in camera viewpoints relative to the subject of interest as the dominant ground plane will have less of a biasing effect.

Fig. 6 depicts an example task from our user interface that was shown to annotators. The first time annotators performed our task they were presented with a short tutorial that included sample images and were instructed on how to use the interface and given feedback when they predicted the incorrect depth ordering. For each task, we also included a short delay before annotators could select their response to encourage them to pay attention to the input image when performing the task. Example annotations from Human3.6M [1] can be seen in Fig. 5. Unsurprisingly, keypoint pairs that have larger relative distances are easier to annotation. For these examples the ground truth accuracies are computed with respect to the corrected ground truth. Example 3D predicted poses on Human3.6M can be seen in Fig. 3.

References

[1] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 2014.

[2] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017.

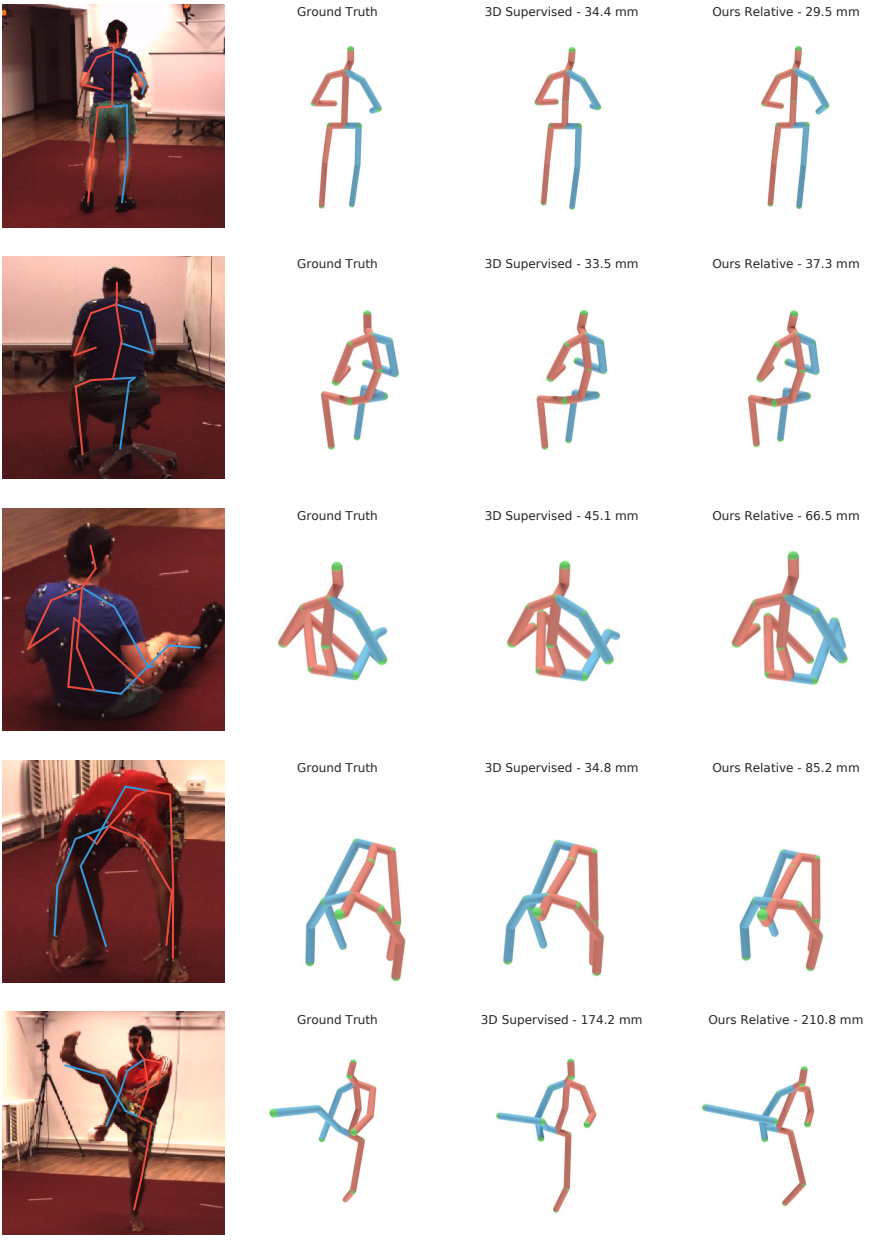
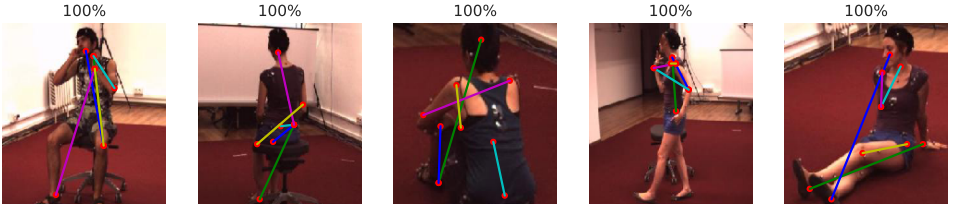
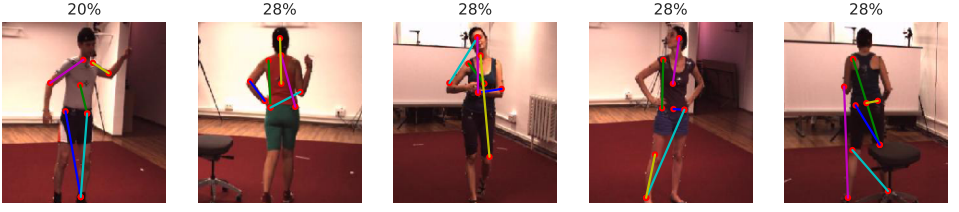


Figure 3: Test time predictions on Human3.6M. Despite using much weaker training data our relative model (Ours Relative 17j GT/GT) produces sensible results for most input poses. Both the supervised and our approach are depicted after rigid alignment, with the pose error displayed on top.

Figure 4: Predicted 3D poses on LSP. Fine-tuning (FT) on LSP significantly improves the quality of our predictions especially for images containing uncommon poses and viewpoints that are not found in Human3.6M.



(a) Easy keypoint pairs.



(b) Difficult keypoint pairs.

Figure 5: Example Human3.6M keypoint pairs. Colored lines link pairs that were shown to annotators. The numbers on top represent the raw accuracy of the crowd provided labels before merging. (a) Easy pairs where no incorrect annotations were made i.e. each of the five annotators annotated all five pairs correctly. (b) Difficult examples where the randomly selected pairs tend to be at a similar distance to the camera, resulting in lower performance.

Figure 6: Our user interface. The annotator's goal is the determine the relative depth of the highlighted keypoints in the left image. The reference image on the right highlights the same keypoints and helps in situations where they are occluded.