

Neural Descent for Visual 3D Human Pose and Shape

Supplementary Material

Andrei Zanfir Eduard Gabriel Bazavan Mihai Zanfir
William T. Freeman Rahul Sukthankar Cristian Sminchisescu

Google Research

{andreiz, egbazavan, mihaiz, wfreeman, sukhthakar, sminchisescu}@google.com

In this supplementary material we provide detail on the CNN feature extraction networks used, the HUND face and hands predictive networks, as well as the way GHUM is used and initialized within a perspective projection camera model, and finally the computation of rotation errors reported in some of the optimization runs in the main paper. We also include video material showing single-image reconstruction – please check it in order to assess the visual 3d reconstruction results qualitatively. N.B. Please observe that the reconstructions are placed plausibly in the 3d scene rather than being shown just overlapped to the image, or in a person-centered coordinate frame.

CNN Semantic Feature Extraction Network. The semantic feature extraction network for 2d keypoints and body parts was trained on a subset of OpenImages which was annotated with 2d keypoints and body part labelling. For augmentation we used translation, cropping, rotations and flipping. As described in the paper we use a ResNet50 [1] backbone. The bottleneck layers of the backbone are passed through deconvolution layers, resized to a common size and concatenated to a final tensor used to store the feature maps \mathbf{F} . The feature maps are used to create separate heads, one for keypoints \mathbf{K} and another one for body parts \mathbf{B} . During training, we first learn the weights for $\mathbf{F} + \mathbf{K}$, then we freeze the weights for \mathbf{F} and learn the weights for \mathbf{B} . While training the weights for the 3d component of **HUND** (second CNN in fig. 1), the weights for \mathbf{F} , \mathbf{K} and \mathbf{B} are frozen.

Perspective Model and Virtual Camera Crops. This section describes our perspective camera model which allows us to plausibly place the GHUM’s 3d reconstructions in the scene (as opposed to just showing them overlaid with input images, or in a person-centered coordinate system). This involves several sub-steps including an adequate initialization for GHUM, as well as the process to properly transfer between the native camera coordinate system where the image was captured and a crop’s coordinate system and associated

virtual camera model of the human detection.

First, we describe the details for computing the initial GHUM state $\mathbf{s}_0 = [\boldsymbol{\theta}_0^\top, \boldsymbol{\beta}_0^\top, \mathbf{r}_0^\top, \mathbf{t}_0^\top]^\top$ (see line 117 in the paper).

We set $\boldsymbol{\theta}_0 = \mathbf{0}$, $\boldsymbol{\beta}_0 = \mathbf{0}$, $\mathbf{r}_0 = [-1, 0, 0, -1, 0, 0]^\top$, values which effectively place the model in an upright position, with the average statistical body shape and pose. For estimating the initial translation \mathbf{t}_0 one needs to take into consideration the camera intrinsics (N.B. this initialization will only be used in order to handle the transformation from the full camera model to a virtual model associated to the crop associated to a human detection).

Our camera model assumes either that camera intrinsics $C = [f_x, f_y, c_x, c_y]^\top$ are known, or default values $f_x = \max(H, W)$, $f_y = \max(H, W)$, $c_x = W/2$, $c_y = H/2$ are used, where H, W are the input dimensions. Given image intrinsics C one has to compute the corresponding crop intrinsics C_c (see line 330 in paper). The crop intrinsics C_c vary depending on the input: we compute an initial translation \mathbf{t}_0 such that the initially reconstructed GHUM mesh projects centrally in the image.

We consider a default set of 2d joint positions \mathbf{J}_d in the crop image which is always of size 480×480 . We compute \mathbf{J}_d by projecting the default model given some default intrinsics $C_d = [480, 480, 240, 240]^\top$ and placing the subject e.g. 2 meters in front of the camera. Since we work with an A-pose, the body joints will all project inside the initial crop and tightly bounded within its borders.

We then minimize the following objective using e.g. least squares and obtain \mathbf{t}_0

$$\mathbf{t}_0 = \arg \min_{\mathbf{t}} \|\mathbf{J}_d - \Pi(\mathbf{J}(\boldsymbol{\theta}_0, \boldsymbol{\beta}_0, \mathbf{r}_0, \mathbf{t}), C_c)\|_2. \quad (1)$$

where $\Pi(\mathbf{J}(\boldsymbol{\theta}_0, \boldsymbol{\beta}_0, \mathbf{r}_0, \mathbf{t}), C_c)$ are the GHUM model joints viewed under a perspective projection using the C_c crop intrinsics.

Rotation angles computation. We describe the details for computing the angles reported in fig. 3. Given rotation

matrices for 2 corresponding joints \mathbf{R}_1 and \mathbf{R}_2 , we compute their difference as $\mathbf{R}_1\mathbf{R}_2^\top$. The angle associated to the rotation difference is then obtained using the formula $\arccos \frac{(\mathbf{R}_1\mathbf{R}_2^\top) - 1}{2}$.

Face and Hands HUND Predictive Networks. We trained the face and hands HUND predictive networks using both fully supervised (FS) and self supervised (SS) regimes. For the fully supervised regime we used data scanned with multiple proprietary systems. The GHUM model was registered to this data and served as pseudo ground truth. Note that the existing large scale datasets (H3.6M and 3DPW) do not have face and hands annotations available. A subset of the registered data was held out and used for testing. For the self supervised regime we used the images from the datasets mentioned in the paper where there was enough resolution for the corresponding body parts. The networks were then trained in a mixed schedule, alternating between self-supervised and fully-supervised batches. For the hands pose prediction networks we report **12.4/13.3mm** MPVPE-PA errors for the left/right hands and **3.4mm** MPVPE-PA errors for the facial expression prediction network. For more results please see the video presented in the supplementary material.

References

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016. 1