GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB

–Supplementary Document–

Franziska Mueller1,2 Florian Bernard1,2 Oleksandr Sotnychenko1,2 Dushyant Mehta1,2 Srinath Sridhar3 Dan Casas4 Christian Theobalt1,2

1 MPI Informatics 2 Saarland Informatics Campus 3 Stanford University 4 Univ. Rey Juan Carlos

In this document we provide details of the RegNet and GeoConGAN networks (Sec. 1), additional quantitative evaluations (Sec. 2), as well as detailed visualizations of our CNN RegNet output and final results (Sec. 3)

1. CNN and GAN Details

1.1. GeoConGAN network

Network Design: The architecture of GeoConGAN is based on the CycleGAN [13], i.e. we train two conditional generator and two discriminator networks for synthetic and real images, respectively. Recently, also methods using only one generator and discriminator for enrichment of synthetic images from unpaired data have been proposed. Shrivastava et al. [9] and Liu et al. [5] both employ an L1 loss between the conditional synthetic input and the generated output (in addition to the common discriminator loss) due to the lack of image pairs. This loss forces the generated image to be similar to the synthetic image in all aspects, i.e. it might hinder the generator in producing realistic outputs if the synthetic data is not already close. Instead, we decided to use the combination of cycle-consistency and geometric consistency loss to enable the generator networks to move farther from the synthetic data thus approaching the distribution of real world data more closely while preserving the pose of the hand. Our GeoConGAN contains ResNet generator and Least Squares PatchGAN discriminator networks.

Training Details: We train GeoConGAN in Tensorflow [1] for 20,000 iterations with a batch size of 8. We initialize the Adam optimizer [4] with a learning rate of 0.0002, β1 = 0.5, and β2 = 0.999.

1.2. RegNet network

Projection Layer ProjLayer: Recent work in 3D body pose estimation has integrated projection layers to leverage 2D-only annotated data for training 3D pose prediction [2]. Since our training dataset provides perfect 3D ground truth, we employ our projection layer merely as refinement module to link the 2D and 3D predictions. We project the intermediate relative 3D joint position prediction using orthographic projection where the origin of the 3D predictions (the middle MCP joint) projects onto the center of the rendered heatmap. Hence, our rendered heatmaps are also relative and not necessarily in pixel-correspondence with the ground truth 2D heatmaps. Therefore, we apply further processing to the rendered heatmaps before feeding them back into the main network branch. Note that the rendered heatmaps are differentiable with respect to the 3D predictions which makes backpropagation of gradients through our ProjLayer possible.

Training Details: We train RegNet in the Caffe [3] framework for 300,000 iterations with a batch size of 32. We use the AdaDelta [12] solver with an initial learning rate of 0.1 which is lowered to 0.01 after 150,000 iterations. All layers which are shared between our network and ResNet50 are initialized with the weights obtained from ImageNet pre-
Figure 2: Qualitative results on the Desk sequence from EgoDexter [6]. RegNet output (rows 1,2) and final tracking.

Figure 3: Qualitative results on community videos from YouTube. RegNet output (rows 1,2) and final tracking.
training [7]. Both, the 2D heatmap loss and the local 3D joint position loss, are formulated using the Euclidean loss with loss weights of 1 and 100, respectively.

**Computational Time:** A forward pass of RegNet in our real-time tracking system takes 13 ms on a GTX 1080 Ti.

### 2. Comparison with RGB-D methods

The 3D tracking of hands in purely RGB images is an extremely challenging problem due to inherent depth ambiguities of monocular RGB images. While our method advances the state-of-the-art of RGB-only hand tracking methods, there is still a gap between RGB-only and RGB-D methods [6, 8, 10]. A quantitative analysis of this accuracy gap is shown in Fig. 1, where we compare our results (dark blue) with the RGB-D method from Sridhar et al. [11] (red).

In order to better understand the source of errors, we perform an additional experiment where we translated the global z-position of our RGB-only results to best match the depth of the ground truth. In Fig. 1 we compare these depth-normalized results (light blue) with our original results (blue). It can be seen that a significant portion of the gap between methods based on RGB and RGB-D is due to inaccuracies in the estimation of the hand root position. Reasons for an inaccurate hand root position include a skeleton that does not perfectly fit the user’s hand (in terms of bone lengths), as well as inaccuracies in the 2D predictions.

### 3. Detailed Qualitative Evaluation

In Figs. 2 and 3 we qualitatively evaluate each of the intermediate stages along our tracking solution as well as the final result. In particular, Fig. 2 shows results on the EgoDexter dataset [6] where a subject grabs different objects in an office environment, and Fig. 3 shows results on community videos downloaded from YouTube. In both figures, we provide visualizations of: heatmap maxima of the 2D joint detections (first row); root-relative 3D joint detections (second row); global 3D tracked hand projected into camera plane (third row); and global 3D tracked hand visualized in a virtual scenario with the original camera frustum (fourth and fifth rows). Please see the supplementary video for complete sequences.

### References


