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

–Supplementary Document–

Franziska Mueller[1,2] Florian Bernard[1,2] Oleksandr Sotnychenko[1,2] Dushyant Mehta[1,2]
Srinath Sridhar[3] Dan Casas[4] Christian Theobalt[1,2]

[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, $\beta_1 = 0.5$, and $\beta_2 = 0.999$.

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-

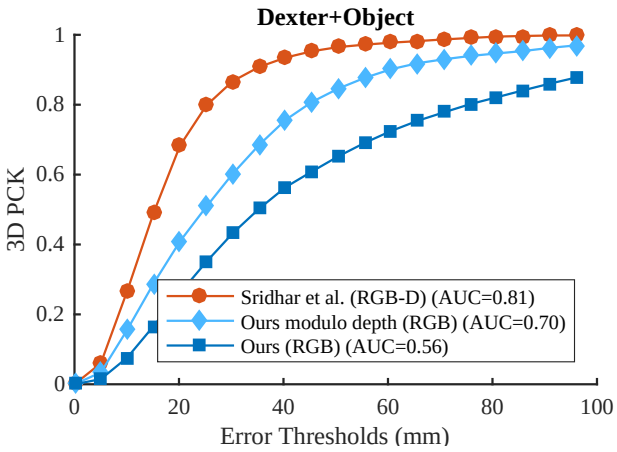

Figure 1: 3D PCK on Dexter+Object. Note that Sridhar *et al*. [11] requires RGB-D input, while we use RGB-only.

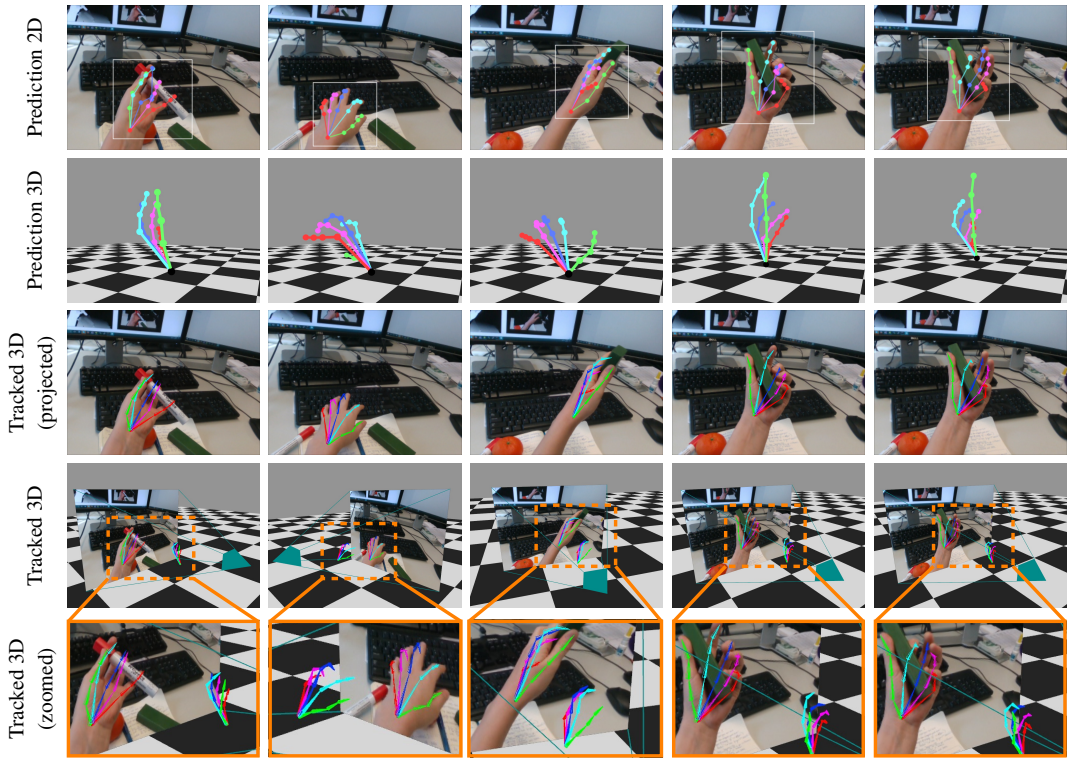

Figure 2: Qualitative results on the *Desk* sequence from EgoDexter [6]. *RegNet* output (rows 1,2) and final tracking.

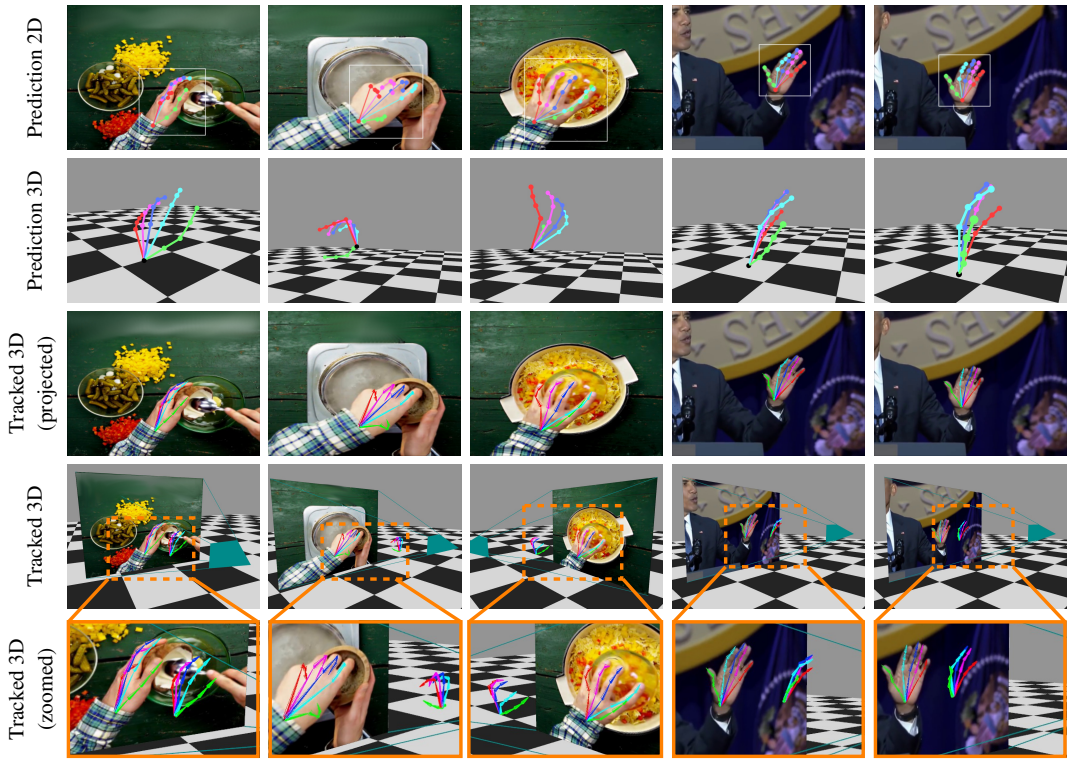

Figure 3: Qualitative results on community videos from YouTube. *RegNet* output (rows 1,2) and final tracking.

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

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] E. Brau and H. Jiang. 3d human pose estimation via deep learning from 2d annotations. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 582–591. IEEE, 2016.

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, 2014.

[4] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] J. Liu and A. Mian. Learning human pose models from synthesized data for robust rgb-d action recognition. *arXiv preprint arXiv:1707.00823*, 2017.

[6] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *International Conference on Computer Vision (ICCV)*, 2017.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[8] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, pages 3633–3642. ACM, 2015.

[9] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and Robust Hand Tracking Using Detection-Guided Optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] S. Sridhar, F. Mueller, M. Zollhöefer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time Joint Tracking of a Hand Manipulating an Object from RGB-D Input. In *European Conference on Computer Vision (ECCV)*, 2016.

[12] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2017.