Supplementary Material for MURAUER: Mapping Unlabeled Real Data for Label AUstERity

Georg Poier Michael Opitz David Schinagl Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
Austria

In this supplemental material we provide more details about our experimental setup and experimental results supporting the claims from the main paper.

1. Details for experimental setup

In the following we describe some more details about our experimental setup. Note that we also make our implementation publicly available[1].

Architecture For the network architectures of the individual modules of our work (*c.f.* Fig. 2 of the main paper), we relied on architectures which have proven successful in related work: The feature extractor f is similar to the model used in [6], *i.e.*, an initial convolutional layer with 32 filters of size 5×5 is followed by a 2×2 max-pooling, four "residual modules" with 64, 128, 256, and 256 filters, respectively, each with five residual blocks [2], and a final fully connected layer with 1024 output units. The pose estimator p consists of two fully connected layers, with 1024 and $3J$ outputs, respectively, where J is the number of predicted joint positions in our case. The mapping layer m is adopted from [8], *i.e.*, it consists of two residual blocks, each with 1024 units. The discriminator h has the same architecture as the mapping m with an additional linear layer to predict a single output. The generator g uses the architecture of the decoder described in [7], which is based on the generator of DCGAN. It consists of four layers of transposed convolutions, each followed by Batch Normalization [3] and a leaky ReLU activation [5]. We add a bilinear upsampling layer prior to the final hyperbolic tangent (tanh) activation in order to upsample from 64×64 to 128×128 in our case.

Optimization For optimization of the model parameters we use Adam [4] with standard parameters, *i.e.*, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We also found it helpful to follow a *warm-up* scheme [1] for the learning rate and decay the learning rate gradually later. More specifically, we start with the learning rate divided by a factor of 10 and multiply it by a factor of 3.33 after the first and the second epoch. Subsequently, the learning rate α_e for epoch e is computed by:

$$\alpha_e = \exp(-\gamma e), \tag{1}$$

where γ determines the speed of the decay and is set to 0.04 in our case. Here, the notion of epoch is always based on the number of real data samples in the dataset (72,757 for the NYU dataset) and independent of the actually used dataset (*e.g.*, sub-sampled real data, synthetic data, *etc.*). That is, the number of iterations per epoch is the same for all experiments (1,137 with a batch size of 64).

Loss weights λ and mini-batch sampling We experimentally found the loss weights used in our work and set $\lambda_c = 0.2$, $\lambda_g = 10^{-4}$ and $\lambda_m = 10^{-5}$. For each mini-batch we independently sample a set of corresponding real and synthetic samples, a set of real samples, a set of synthetic samples and a set of unlabeled samples such that there is an equal number of samples from each of the four sets (*i.e.*, 16 samples per set in our case).

Data augmentation We used online data augmentation. That is, each time we sample a specific image we also sample new transformation parameters. In this work we randomly rotate the loaded image, randomly sample the location of the crop and add white noise to the depth values. The rotation angle is uniformly sampled from $[-60°, 60°]$ and the location offset as well as white noise is sampled from a normal distribution with $\sigma = 5\,\mathrm{mm}$.

2. Full NYU dataset for view prediction

In our recent work [7], in which we showed that the view prediction objective is a good proxy for pose specifity

[1] https://poier.github.io/murauer

Dataset (n)	Mean absolute error (MAE)
NYU-CS (43,640)	0.123
NYU (72,756)	0.107

Table 1: **View prediction with additional data.** The results of view prediction trained on the NYU-CS subset as used in [7] and on the full NYU set by using the camera views with roughly the same viewpoints for each frame. See text for details.

we decided to leave out about 40% of the NYU dataset for learning to predict different views because the camera setup has been changed when capturing this part. As briefly mentioned in the main paper, we found that the camera views in the left-out part, which are closest to the view points we used for the NYU-CS in [7] are roughly the same for at least one camera throughout the whole dataset. That is, ignoring slight changes of the camera poses, we could employ the whole NYU dataset, despite assuming a fixed setup as in [7].

View prediction Here, we show that a model trained for view prediction can exploit the full set, despite the slight changes of the setup. To this end, we train on the reduced NYU-CS set [7] or the full NYU dataset, respectively, and compared the results on the standard test set. In our experiments the Mean absolute error (MAE) is reduced by 13% (see Tab. 1) when exploiting the full NYU dataset instead of the reduced NYU-CS. This shows that even for the base task of view prediction the additional data can be exploited despite the slightly changed camera poses.

Semi-supervised hand pose estimation Furthermore, we evaluate the model we used in [7] using the full dataset directly on hand pose estimation. We investigate how the error evolves with a gradually increased number of labeled samples. We gradually increase the number of labeled subsets of the NYU-CS dataset up to the ∼44k samples from NYU-CS and compare to the result when using all (∼73k) samples in Fig. 1. We can see that the error starts to level up when using subsets from NYU-CS, but experiences a sudden drop when exploiting the additional data. The fact that the results do not just improve gradually – as it would probably be the case if we would provide "more of the same" data, *i.e.*, a denser sampling of the existing data – again indicates that the model can indeed exploit the additional data included in the full set.

3. Qualitative analysis

Domain gap in latent space In the main paper we showed a t-SNE visualization of the latent representation

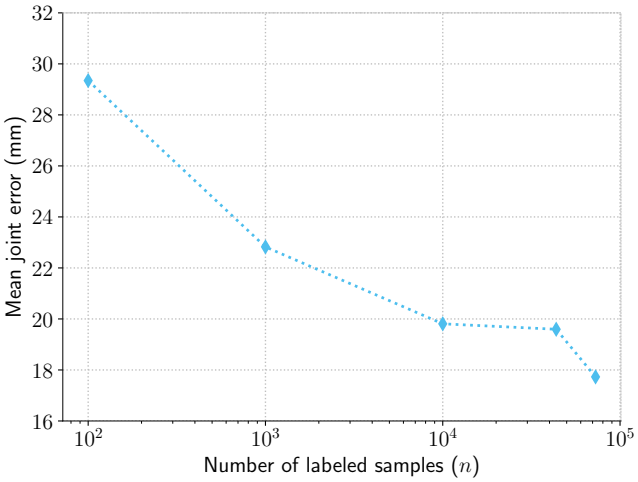
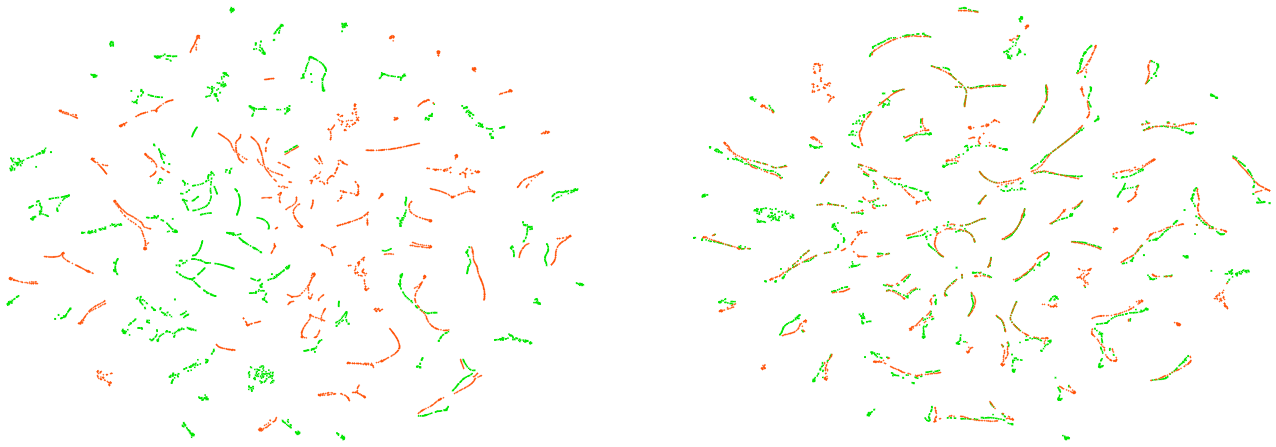


Figure 1: **Semi-supervised learning with additional data.** Results of the semi-supervised method introduced in [7] when evaluated on the NYU-CS subset [7] with ∼44k samples (*i.e.*, leftmost four experiments in plot) and when exploiting the full NYU set with ∼73k samples. See text for details.

learned with our method. In a similar manner, here, we want to qualitatively illustrate the importance of tackling the domain gap between synthetic and real data when exploiting synthetic data for hand pose estimation. To this end, we compare the t-SNE visualization of the latent space learned solely with synthetic data to the the visualization of the representation obtained with our method in Fig. 2. Despite that the poses of real and synthetic images are corresponding, we can see that the samples from the two domains take up rather different areas in the visualization for the model trained only on synthetic data (see Fig. 2a). Whereas, the visualization for our method, which also uses unlabeled data and only 100 labeled real samples, shows that the real and synthetic data is well aligned (see Fig. 2b).

Error cases We analyze the error cases for our model trained with 100 labeled real samples. Representative samples from the 100 frames with largest mean error are shown in Fig. 3, samples from the frames with largest maximum error are shown in Fig. 4. We find that our model has problems especially if none of the fingers is clearly visible in the depth frame, *i.e.*, the frame has a "blob-like" appearance.

For the frames our model had the largest problems with, we search the nearest neighbors in the training set. We find the nearest neighbors based on the average joint distance between the corresponding ground truth annotations (after shifting the annotations to the origin). Fig. 5 shows the nearest neighbors for some selected test samples. We find that for some samples there are no close nearest neighbors in the training set, and we hypothesize that for such "blob-like"



(a) (b)

Figure 2: **Visualization of latent representations.** The latent representations of corresponding real (green) and synthetic (orange) samples visualized using t-SNE [9]. Visualization for a model trained on synthetic data (a) and our model trained on synthetic, unlabeled and 0.1% of labeled real samples (b). Best viewed in color.

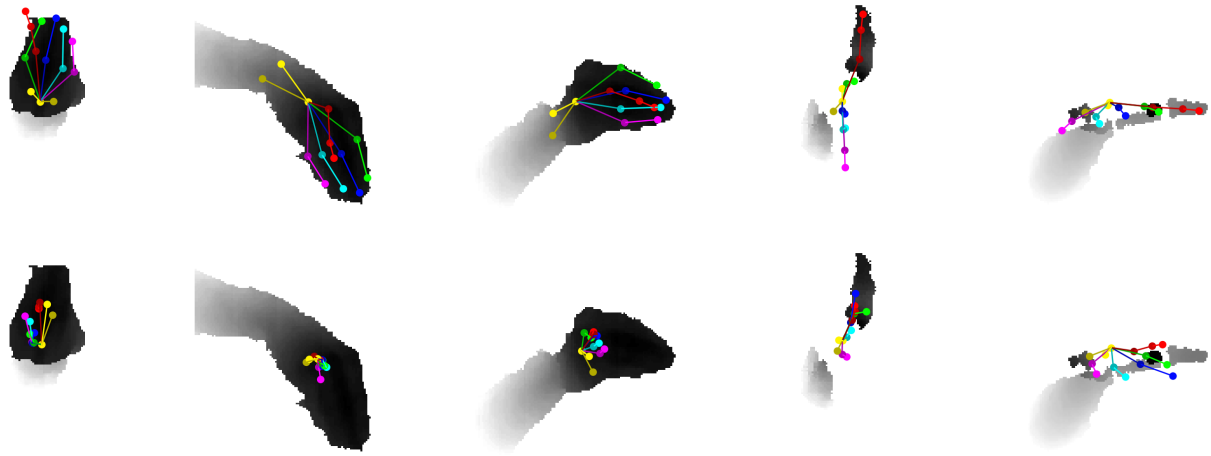


Figure 3: **Frames with largest mean error.** Test samples overlaid with ground truth (top row) and the predictions of our model (bottom row). Note, 90 of the 100 frames with the largest mean error are variations of the leftmost three frames.

structures it is especially difficult to obtain valuable feedback from the view prediction objective. Also note, that the model we are analyzing was trained on only 100 labeled real samples and the labels for the nearest neighbors shown in Fig. 5 were not used.

References

[1] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch SGD: training imagenet in 1 hour. *ArXiv e-prints*, abs/1706.02677, 2017. 1

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 1

[3] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015. 1

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015. 1

[5] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML Workshops*, 2013. 1

[6] M. Oberweger and V. Lepetit. DeepPrior++: Improving fast and accurate 3d hand pose estimation. In *Proc. ICCV Workshops*, 2017. 1

[7] G. Poier, D. Schinagl, and H. Bischof. Learning pose specific representations by predicting different views. In *Proc. CVPR*, 2018. 1, 2

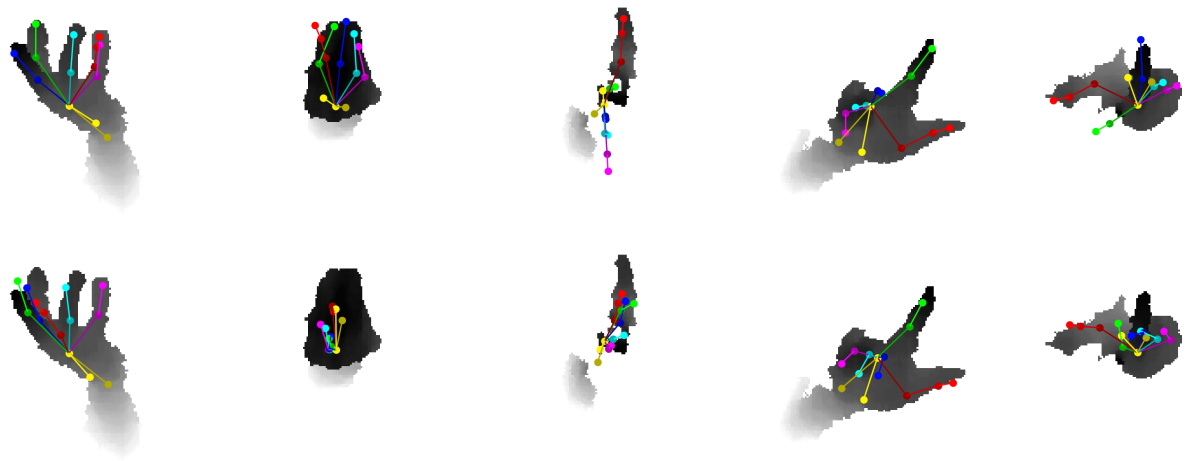


Figure 4: **Frames with largest maximum error.** Test samples overlaid with ground truth (top row) and the predictions of our model (bottom row). The errors are mainly due to strongly distorted samples and annotation errors. In this case, 79 of the 100 frames with largest maximum error are variations of the three leftmost frames.

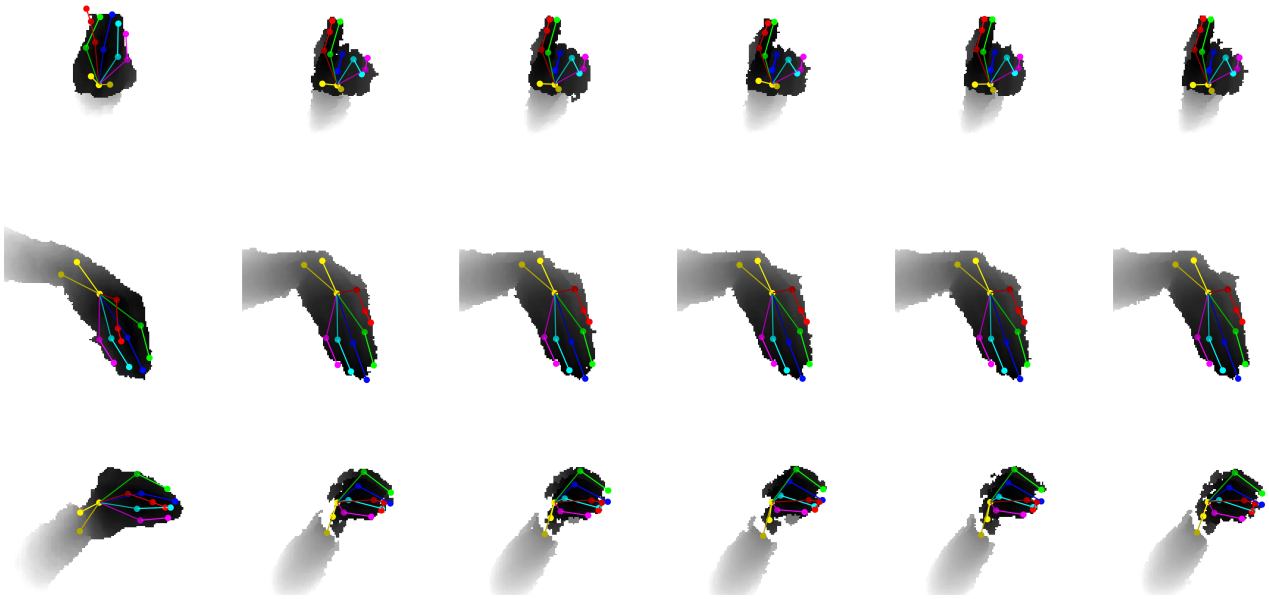


Figure 5: **Nearest neighbors in training set.** The test samples with largest error (*c.f.*, Fig. 3 and Fig. 4) and their nearest neighbors in the training set. Leftmost column shows the test sample, the remaining columns show the corresponding nearest neighbors from the training set. Note, the training samples were used only unlabeled.

[8] M. Rad, M. Oberweger, and V. Lepetit. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. In *Proc. CVPR*, 2018. 1

[9] L. van der Maaten and G. E. Hinton. Visualizing data using t-SNE. *JMLR*, 9:2579–2605, 2008. 3