Learning to Train with Synthetic Humans **Supplementary Material**

David T. Hoffmann[1], Dimitrios Tzionas[1], Michael J. Black[1], and Siyu Tang[1,2,3]

[1] Max Planck Institute for Intelligent Systems, Germany
[2] University of Tübingen, Germany
[3] ETH Zürich
dhoffmann, dtzionas, black, stang@tuebingen.mpg.de

1 Data Generation

MoCap Data. To pose the body we use 1,515 MoCap sequences from the CMU dataset [1], 62 from HumanEva [14], 41 from the PosePrior dataset [3] and 157 unpublished sequences recorded with our own motion capture system. To reduce the similarity of poses we subsample every 10th frame, resulting in 12 fps for most of the datasets. We end up with 253,762 individual poses.

Hand Poses. Without variations of hand poses, a keypoint detector trained on the synthetic data might not generalize to other hand poses. To avoid this, we use SMPL+H [13] and pose the hands and fingers. However, conventional MoCap systems do not record the pose of fingers [1, 3, 14]. To obtain realistic poses for hands and fingers we use the "embodied hands" dataset [13].

Shape. Besides body pose, humans differ in their body shape. To express these differences in our synthetic datasets we extract shape parameters β from standard MoCap datasets using MoSh [9].

Textures. Textures have a large influence on the perceived realism of synthetic data. For the synthetic humans we use the textures published with the SURREAL dataset [18]. The dataset provides 772 textures of people in casual clothing and 157 in minimal clothing. The former were collected by the authors of [18]. The latter were acquired from the CAESAR dataset [11]. We use 80% of these textures and keep the remaining 20% for validation and test data, even though we do not use any synthetic validation or test data for this particular work. The reason for this decision is to keep the data splits for training, test and validation consistent across projects.

Noise and Lighting. Real images are subject to multiple sources of noise, for example motion blur or defocus. Furthermore, they are taken under varying lighting conditions. We model noise with Gaussian blur. We blur x and y direction independently with a probability of 0.5, each. The size of the Gaussian kernel in pixels is given by the absolute value sampled from a standard normal distribution. We vary lighting conditions similarly as [18, 10] using spherical harmonics [6] with randomly drawn coefficients.

Sampling Motions. Sampling MoCap sequences randomly does not guarantee that every MoCap sequence is used. To ensure that, we select the sequence

for the first synthetic human deterministically. MoCap sequences for all other synthetic humans are sampled randomly. The probability of sampling a MoCap sequence S_j is given by $p_j = \frac{|S_j|}{\sum_{i=1}^{|S|} |S_i|}$. Here $|S_i|$ denotes the number of frames of sequence i and $|S|$ the number of sequences. Since the chosen sequences might have different length, we further randomly sample for each one the frame ID for the starting frame, to encourage the use of the whole sequences.

Changing Rendering Parameters. To increase variance in the dataset multiple parameters are changed. For each MoCap sequence we change the camera position, number of humans and the global rotation of each synthetic human. In contrast, background image and lighting as well as position, pose, shape and texture of synthetic humans are changed for each frame.

Scene Generation. After sampling all these parameters, each virtual human is placed on an invisible ground plane in the field of view of the camera. Maximal distance to the camera are 12 m.

Collision Detection. The meshes of synthetic humans in the 3D scene might intersect, resulting in physically impossible configurations. A quick and easy way of detecting collisions are axis aligned bounding boxes. However, their usage results in a large number of false positives and limits the distances between virtual humans. For multi-person pose estimation, however, small distances between humans are frequent and should be represented in the training data. We draw inspiration from [17] and use bounding volume hierarchies (BVH) [16] instead. This is an efficient method to check for collisions on triangle level. Thus, it allows for smallest possible distances between the meshes.

Whenever two synthetic humans collide, we search for a new, valid position for one of them before rendering the image. Frames with mesh self-collisions are not rejected, as they are very frequent for highly articulated poses, which might be important for a pose-estimation dataset.

2 Datasets

Tab. A.1 shows a comparison of our datasets to related other datasets. JTA and SURREAL are much larger, however SURREAL only considers a single person in indoor environments and JTA only urban scenes in surveillance scenarios. Our datasets have a much higher variety in terms of scenes.

Example images for $\mathcal{D}_S$ can be seen in Fig. A.1

2.1 $\mathcal{D}_M$ and $\mathcal{D}_{Style}$ - Qualitative Results

Fig. A.2 and A.3 show example images of $\mathcal{D}_M$ and the corresponding images from $\mathcal{D}_{Style}$. In particular Fig. A.2 (A) shows that the style transfer method generates realistic variations of textures and changes their color. In Fig. A.2 (B, C) and Fig. A.3 (A, B, C) it can be seen that the method adapts the lighting of textures to fit better into the scene. It greatly improves blending-in of synthetic humans. Fig A.2 (D) and Fig A.3 (D) show failure cases. Here, larger parts of the background were included in the human mask of mask-RCNN [7, 2]. As a

Fig. A.1. Example images from $\mathcal{D}_S$. The dataset contains images with interesting poses, challenging backgrounds, variance in camera position and heavy occlusion.

Table A.1. Number of frames and poses for our datasets and for the most related synthetic datasets. All datasets are multi-person datasets, except for SURREAL. For $\mathcal{D}_M$ and $\mathcal{D}_{Style}$ "#Synthetic Humans" corresponds to the additional number of synthetic humans. For $\mathcal{D}_S$, SURREAL, JTA and SURREAL-style multi-person "#Synthetic Humans" corresponds to the number of annotated poses.

Dataset		#frames	#Synthetic Humans
$\mathcal{D}_S$	ours	70,379	580,693
$\mathcal{D}_M$	ours	74,628	279,605
$\mathcal{D}_{Style}$	ours	74,352	279,278
SURREAL	[18]	6,536,752	6,536,752
JTA	[5]	460,800	10,000,000
SURREAL-style multi-person	[12]	40,000	186,000

result, synthetic humans are partially stylized in the style of the background. This leads to the observable ghost-effect.

3 Training

Pose Estimation Network. Unfortunately not all hyper-parameters used for training on the MPII pose estimation dataset were provided for the OpenPose network. Our hyper-parameter search results in a model that approaches the original performance. Differences in performance may be due to a different choice of optimizer. We use the more common Adam algorithm [8], whereas Cao et al. [4] rely on SGD. Our hyperparameter search lead to a learning rate of $lr = 0.0001$, $\beta_1 = 0.8$ and $\beta_2 = 0.999$. Furthermore, we found that a learning rate decay improves results. We decay the learning rate every 20.000 steps with a decay rate of 0.66. We use a batch size of 32.

Most of our models are initialized with weights pretrained on real data. The only 2 models that are not pretrained on real pose estimation data are $\mathcal{M}_{\mathcal{D}_R}$ and $\mathcal{M}_{\mathcal{D}_S}$. For these models we follow the same procedure as proposed by Cao et al. [4] and initialize them with the first 10 layers of VGG-19. The remaining weights are randomly initialized.

Training with Synthetic Data. All models trained with synthetic data are initialized with pretrained weights. These weights are obtained by training OpenPose for 70.000 steps on real data. Whenever we start from these pretrained weights, we use an initial learning rate of $lr = 0.00005$.

3.1 Data Augmentation

To increase the number of training samples we apply standard data augmentation techniques. We implement the same data augmentation pipeline as Cao et al. [4], however the hyperparameters might be different. We scale the image in the range $[0.4, 1.6]$, rotate it by a uniformly sampled value in the interval $[-45°, 45°]$.

Fig. A.2. Example images from $\mathcal{D}_M$ and the respective image from $\mathcal{D}_{Style}$. Last row shows a failure case with ghost-like appearance of synthetic humans.

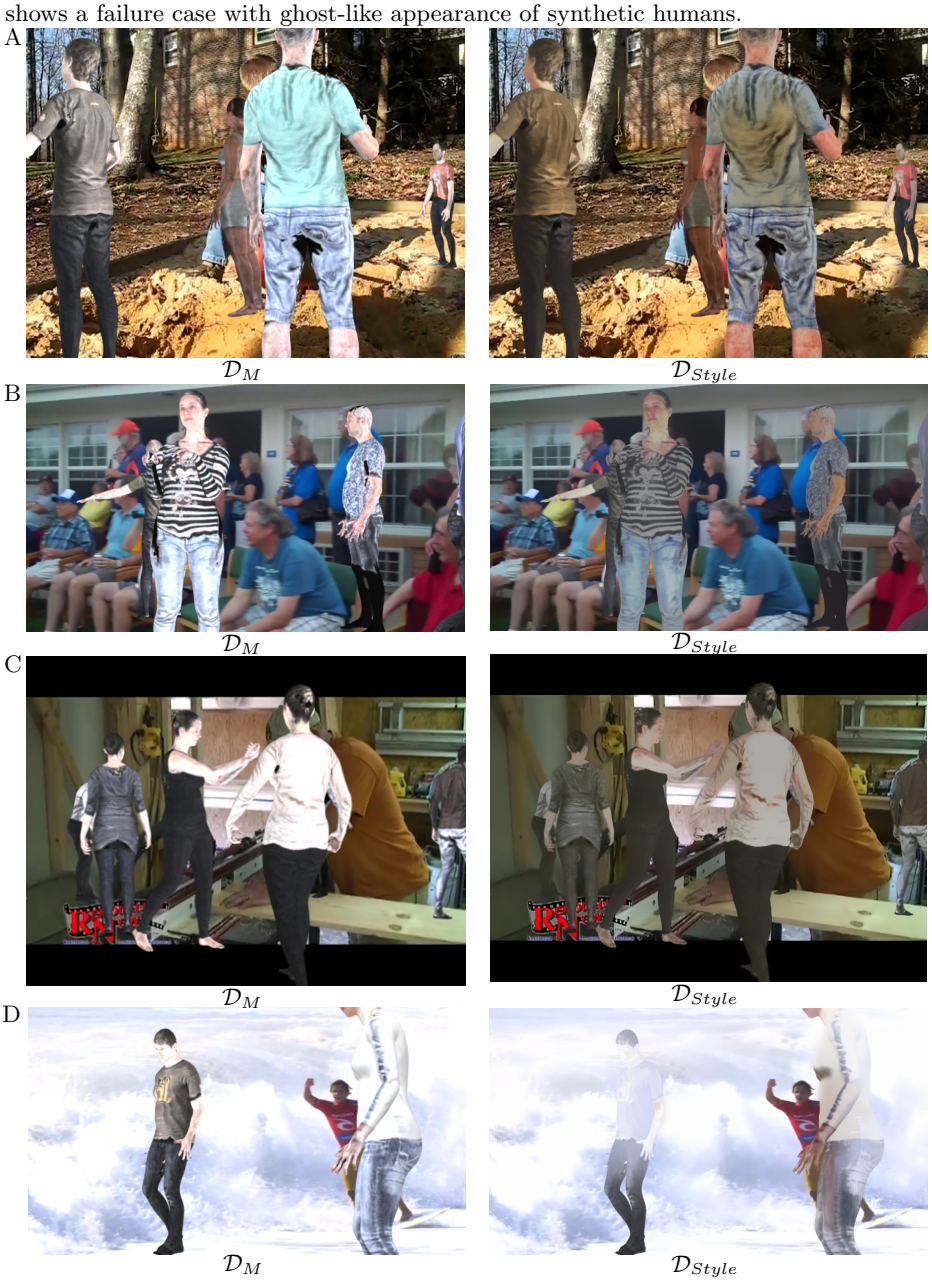


Fig. A.3. Example images from $\mathcal{D}_M$ and the respective image from $\mathcal{D}_{Style}$. Last row shows a failure case with ghost-like appearance of synthetic humans.

Furthermore, each image is cropped around a target person. Size of the crop is 368×368 px. We add noise to the center of the crop. It is uniformly sampled from $[-50, 50]$ px. Finally, with probability of 0.5 the image is flipped horizontally.

3.2 The Teacher Network

To optimize the teacher we use a constant learning rate of 0.00005. Similar to optimization of the OpenPose network we use the Adam algorithm with $\beta_1 = 0.8$ and $\beta_2 = 0.999$.

Architecture. The teacher network consists of two parts. The first part are the first 10 layers of VGG-19 [15] and is identical to the OpenPose feature extractor. Similar to the OpenPose feature extractor the weights of VGG-19 are used to initialize it. The second part of the teacher network differs from the OpenPose network. It consists of $3 \times 3 \times 256$ and $3 \times 3 \times 128$ convolutional layers, followed by two $3 \times 3 \times 64$ layers. After every two layers we add a max pooling layer. The last two layers are fully-connected with 512 units each.

4 Multi-Person Pose Estimation - Qualitative Results

Qualitative results for $\mathcal{M}_{\mathcal{D}_R}$, "adversarial Teacher" C and $\mathcal{M}_{\mathcal{D}_R+\mathcal{D}_{Style}+\text{masks}}$ can be seen in Fig. A.4, Fig. A.5 and Fig. A.6. While a clear improvement can be seen by "adversarial Teacher" C and $\mathcal{M}_{\mathcal{D}_R+\mathcal{D}_{Style}+\text{masks}}$ over $\mathcal{M}_{\mathcal{D}_R}$, differences between "adversarial Teacher" C and $\mathcal{M}_{\mathcal{D}_R+\mathcal{D}_{Style}+\text{masks}}$ are more subtle. We found that their predictions are qualitatively on par for most images.

As can be seen in Fig. A.4 (A, B) and Fig. A.6 (A) grouping of joints fails frequently for $\mathcal{M}_{\mathcal{D}_R}$. In contrast, erroneous grouping can be less frequently observed for our models. Thus, the models trained on our datasets improve in their ability to group joints. For very crowded scenes as shown in Fig. A.4 (A) and Fig. A.6 (A) the "adversarial Teacher" C seems to outperform $\mathcal{M}_{\mathcal{D}_R+\mathcal{D}_{Style}+\text{masks}}$. Thus, adversarial training on the more crowded purely synthetic dataset is beneficial for real crowded scenes.

Besides grouping, training on our datasets seems to improve the detection of occluded people and occluded keypoints. Examples for this can be seen in Fig. A.4 (A, B, C), Fig. A.5 (B, C) and Fig. A.6 (A). Of particular interest is Fig. A.5 (A). Here it can be seen that the training on purely synthetic data with adversarial teacher leads to better detection of people under challenging imaging conditions and improves the detection and grouping of joints in front of highly cluttered backgrounds.

Last, Fig. A.6 (A, B) suggest that "adversarial Teacher" C improves the prediction for uncommon camera positions. Thus, training on the most challenging camera positions improves the predictions for such images in comparison to $\mathcal{M}_{\mathcal{D}_R}$ and $\mathcal{M}_{\mathcal{D}_R+\mathcal{D}_{Style}+\text{masks}}$.

Fig. A.4. Example images with detected poses for $\mathcal{M}_{\mathcal{D}_R}$, the model trained with adversarial teacher using the camera pitch grouping ("adversarial Teacher" C) and our best model $\mathcal{M}_{\mathcal{D}_R+\mathcal{D}_{Style}+\text{masks}}$.

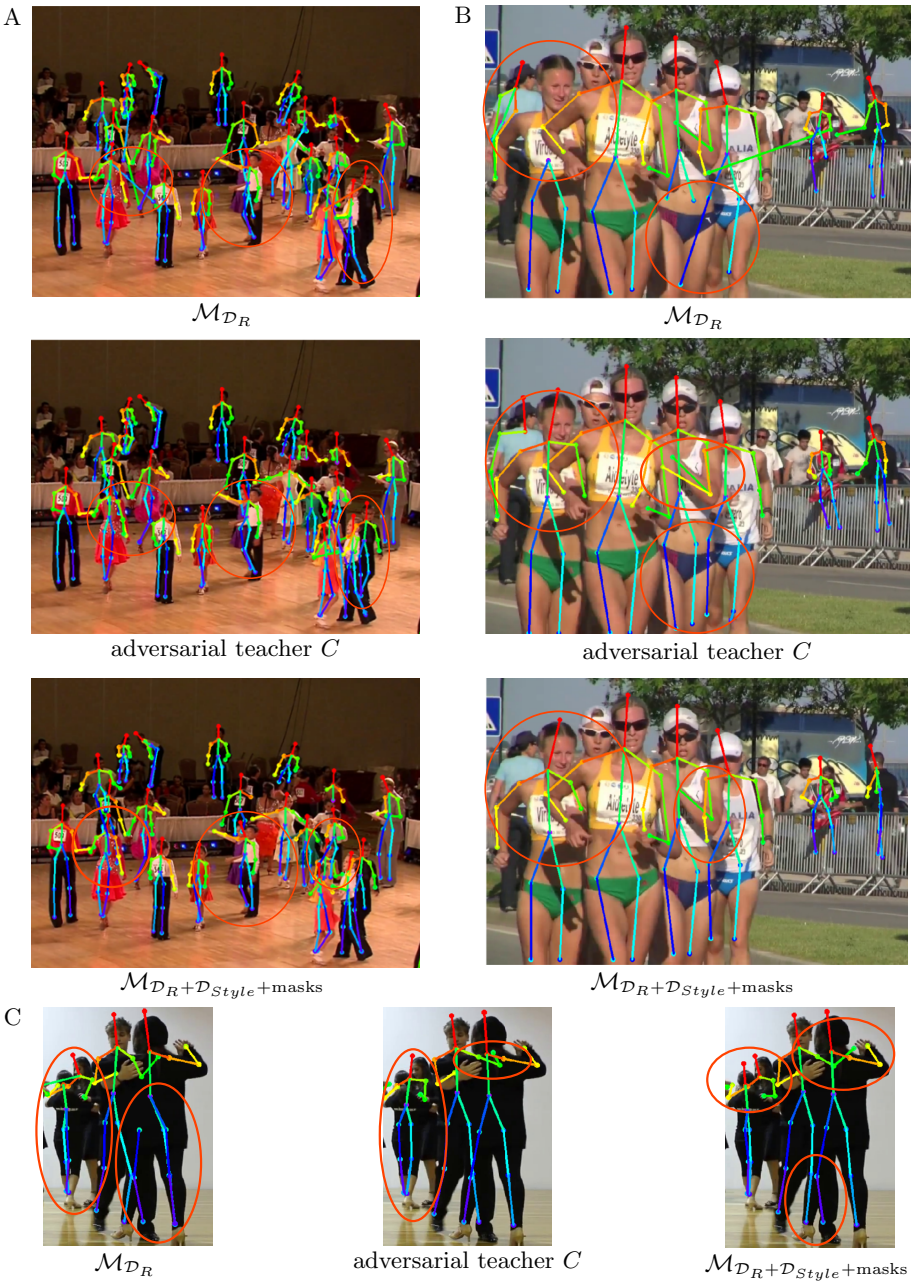
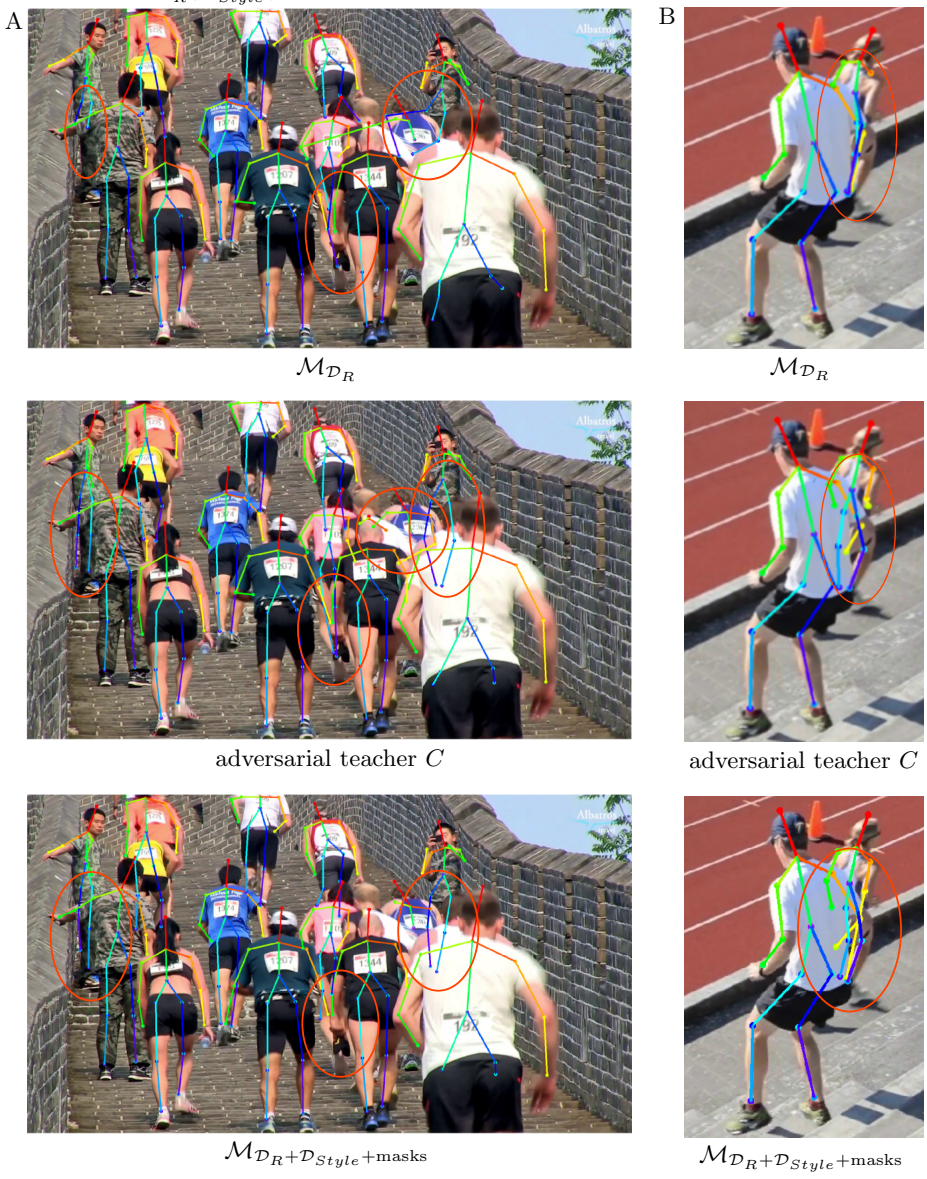


Fig. A.5. Example images with detected poses for $\mathcal{M}_{\mathcal{D}_R}$, the model trained with adversarial teacher using the camera pitch grouping ("adversarial Teacher" C) and our best model $\mathcal{M}_{\mathcal{D}_R+\mathcal{D}_{Style}+\text{masks}}$.

Fig. A.6. Example images with detected poses for $\mathcal{M}_{\mathcal{D}_R}$, the model trained with adversarial teacher using the camera pitch grouping ("adversarial Teacher" C) and our best model $\mathcal{M}_{\mathcal{D}_R+\mathcal{D}_{Style}+\text{masks}}$.



References

1. Carnegie-mellon mocap database. http://mocap.cs.cmu.edu
2. Abdulla, W.: Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN (2017)
3. Akhter, I., Black, M.J.: Pose-conditioned joint angle limits for 3d human pose reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1446–1455 (2015)
4. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1302–1310. IEEE (2017)
5. Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., Cucchiara, R.: Learning to detect and track visible and occluded body joints in a virtual world. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 430–446 (2018)
6. Green, R.: Spherical Harmonic Lighting: The Gritty Details. Archives of the Game Developers Conference (Mar 2003), http://www.research.scea.com/gdc2003/spherical-harmonic-lighting.pdf
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2980–2988. IEEE (2017)
8. Kinga, D., Adam, J.B.: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR). vol. 5 (2015)
9. Loper, M., Mahmood, N., Black, M.J.: Mosh: Motion and shape capture from sparse markers. ACM Transactions on Graphics (TOG) **33**(6), 220 (2014)
10. Ranjan, A., Romero, J., Black, M.J.: Learning human optical flow. In: 29th British Machine Vision Conference (Sep 2018)
11. Robinette, K.M., Blackwell, S., Daanen, H., Boehmer, M., Fleming, S.: Civilian american and european surface anthropometry resource (caesar), final report. volume 1. summary. Tech. rep., DTIC Document (2002)
12. Rogez, G., Weinzaepfel, P., Schmid, C.: Lcr-net++: Multi-person 2d and 3d pose detection in natural images. IEEE transactions on pattern analysis and machine intelligence (2019)
13. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) **36**(6) (Nov 2017)
14. Sigal, L., Balan, A.O., Black, M.J.: Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. International journal of computer vision **87**(1-2), 4 (2010)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Teschner, M., Kimmerle, S., Heidelberger, B., Zachmann, G., Raghupathi, L., Fuhrmann, A., Cani, M.P., Faure, F., Magnenat-Thalmann, N., Strasser, W., Volino, P.: Collision detection for deformable objects. In: Eurographics. pp. 119–139 (2004)
17. Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., Gall, J.: Capturing hands in action using discriminative salient points and physics simulation. International Journal of Computer Vision (IJCV) **118**(2), 172–193 (Jun 2016), https://doi.org/10.1007/s11263-016-0895-4
18. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: CVPR (2017)