Adapting Egocentric Visual Hand Pose Estimation Towards a Robot-Controlled Exoskeleton

Gerald Baulig, Thomas Gulde, and Cristóbal Curio

Computer Science, Reutlingen University, Reutlingen, Germany
http://cogsys.reutlingen-university.de
{thomas.gulde,cristobal.curio}@reutlingen-university.de

Abstract. The basic idea behind a wearable robotic grasp assistance system is to support people that suffer from severe motor impairments in daily activities. Such a system needs to act mostly autonomously and according to the user's intent. Vision-based hand pose estimation could be an integral part of a larger control and assistance framework. In this paper we evaluate the performance of egocentric monocular hand pose estimation for a robot-controlled hand exoskeleton in a simulation. For hand pose estimation we adopt a Convolutional Neural Network (CNN). We train and evaluate this network with computer graphics, created by our own data generator. In order to guide further design decisions we focus in our experiments on two egocentric camera viewpoints tested on synthetic data with the help of a 3D-scanned hand model, with and without an exoskeleton attached to it. We observe that hand pose estimation with a wrist-mounted camera performs more accurate than with a head-mounted camera in the context of our simulation. Further, a grasp assistance system attached to the hand alters visual appearance and can improve hand pose estimation. Our experiment provides useful insights for the integration of sensors into a context sensitive analysis framework for intelligent assistance.

Keywords: Hand Pose Estimation, Egocentric View, Grasp Assistance, Simulation

1 Introduction

Neurorobotics is a promising field to potentially support patients suffering from debilitating conditions, e.g. stroke, and allow them to regain motoric functions. (Fig. 1). Soekadar *et al.* [23] present an exoskeleton-based robotic system to support and treat patients suffering from motor impairments based on electroencephalography (EEG) and electrooculography (EOG) input. The concept is extendable to integrate intelligent sensors that perceive and understand the scene and can potentially increase the usability of such or similar systems for daily life usage. Such an approach would understand the context of the grasp interaction and the user's intention for a more autonomous assistance, since the

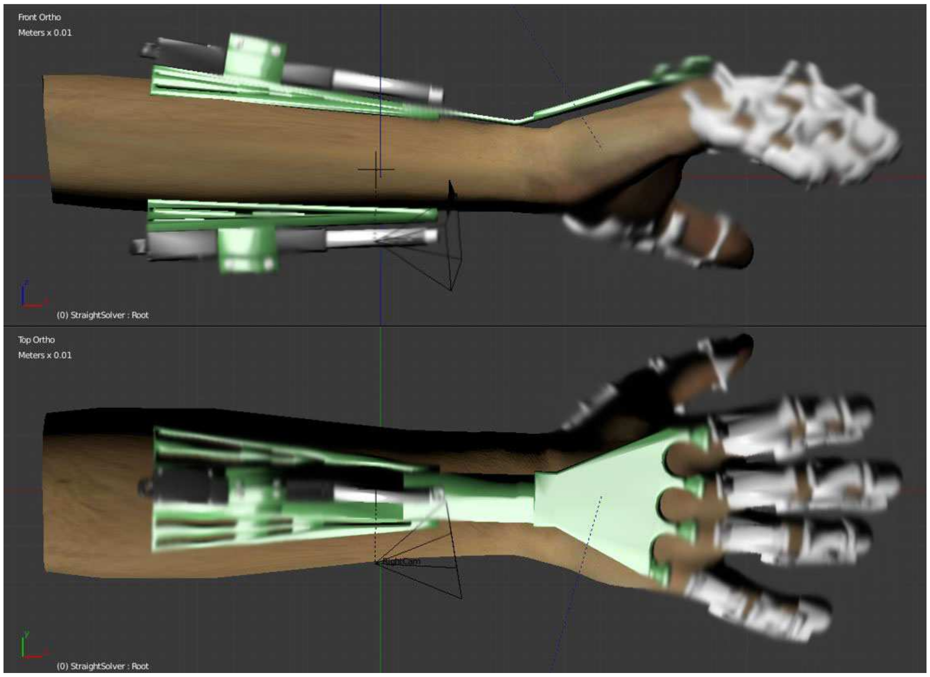

Fig. 1. Design of an exoskeleton prototype for grasp assistance simulated on our 3D hand scan model.

interpretation of user input, solely based on EEG or EOG, might be prone to error. A computer vision based control instance could provide valuable input to support the decision whether the hand is or should be in a certain state. Therefore an autonomous grasp assistance could profit from hand pose estimation to control the grasping process and intervene, if necessary, to reach a certain goal. Parts of a grasp interaction are the exoskeleton, graspable objects as well as the bare hand. The development of a perception system would need to consider challenges like the estimation of the pose of the exoskeleton, the pose of the bare hand, the shape and pose of the graspable object as well as the classification of the interaction as input for a higher-level micro-controller.

In this paper we focus on comparing hand pose estimation with and without an exoskeleton covering the hand and take a closer look at appropriate camera setups for further design decisions of the overall exoskeleton system. Thus our guiding questions are how does a computer vision based hand pose estimation perform on our exoskeleton and what are appropriate egocentric viewpoints for the camera sensors? Since the prototype is not available yet, we run a closed test scenario with our 3D test framework. Here we demonstrate our test framework with the evaluation of two egocentric sensor viewpoint positions, on the forehead and on the wrist (Fig. 2). We present a closed test framework to compare several setups and scenarios to aid further design decisions for the exoskeleton. Although

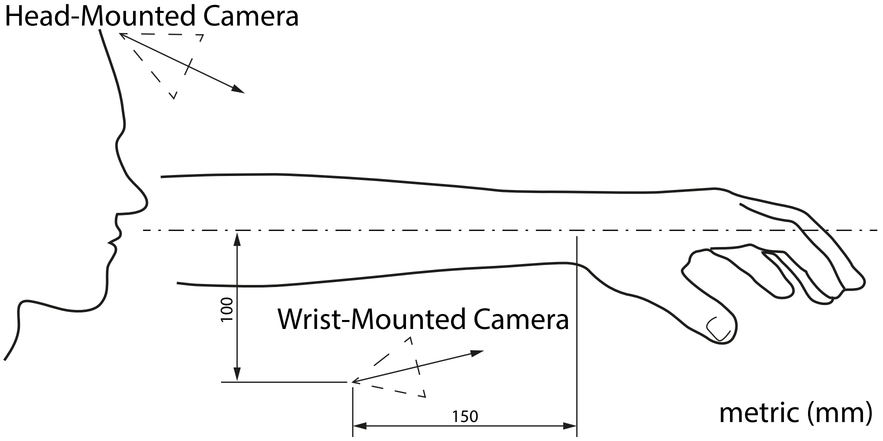

Fig. 2. The two camera setups we demonstrate and evaluate in a closed test scenario.

explicit knowledge about the actuator state of an exoskeleton may support the estimation accuracy, this closed test framework is focusing on a solution with computer vision based on deep learning only. We want to point out that our test framework is based on an established method for hand pose estimation. The system can support studying generalization from synthetic data to real scenarios in the wild.

The paper is structured as follows. In section 3.1 we illustrate our own synthetic data generator to create data for monocular hand pose estimation especially in the context of a wearable exoskeleton allowing for flexible egocentric sensor viewpoint selection. For the hand pose estimation we employ an approach based on Convolutional Neural Networks (CNN) as described in section 3.2. In section 4 we train and evaluate our network with the produced simulated data. Further, in section 5, we compare the accuracy of pose estimation between a head- versus wrist-mounted camera with the two conditions of a bare hand versus a hand covered by an exoskeleton. Based on a quantitative evaluation and the illustration of qualitative examples of our own and others' datasets we discuss our camera setups in section 6. We conclude in section 7 and outline potential next steps.

2 Related Work

Hand pose estimation is a very active field of research, documented in reviews like [8, 17, 3]. Primarily, recent research has focused on joint-based hand pose estimation, as listed in Chen *et al's.* project [5]. Only few works pursue an approach based on egocentric perception viewpoint. We adopt the idea of Chan *et al.* [4] to mount cameras on the head or wrist. The authors present a multiple wearable camera setup for egocentric activity recognition. This framework provides a scene- and grasp classification on two synchronized image streams. Their image-based grasp classification has an accuracy of about 51% for a head-mounted camera, but a wrist-mounted camera performs 5.5% better. To the best of our knowledge, a wrist-mounted camera setup has not yet been tested for joint-based hand pose estimation. These estimated joint positions may offer a new representation for further high-level classification steps.

Rogez *et al.* [20] employ Random Forest Trees on images of a chest-mounted depth-camera for hand pose estimation and classify up to 71 hand poses in [19]. First, the class of the pose gets estimated, then a preset pose configuration gets aligned onto a reference image. Bambach *et al.* [2] employ a head-mounted camera like *Google Glass* and a CNN for hand segmentation of 4 classes (`my-left-hand`, `my-right-hand`, `your-left-hand` and `your-right-hand`). Their dataset *EgoHands* provides scenes of two persons sitting in front of each other and playing cards or other games. Mueller *et al.* [16] employ chest- or rather shoulder-mounted depth-cameras and two derivations of *ResNet50* [10] to first detect the hand and then estimate the joint positions. After the joints are estimated, a kinematic model is aligned to these points. Their dataset *EgoDexter* provides annotations for visible fingertips only.

Maekawa *et al.* [14] employ sensor fusion of a wrist-mounted camera, microphone, accelerometer, illuminometer and magnetometer to classify with a hidden Markov model the activity of the hand, but do not focus on estimation of its pose. A wrist-mounted setup with pose-estimation is presented by Kim *et al.* [13]. They employ an infrared (IR)-camera combined with IR-laser. The IR-laser generates a kind of structured light as a line crossing the proximal bones of the hand. Those handcrafted features and the depth estimation of the fingertips are used to estimate the pose of the forward kinematic chain of each finger relative to the calibrated camera.

In this paper we want to evaluate the performance of a state-of-the-art CNN-based hand pose estimation method for wrist-mounted RGB-camera images compared to head-mounted RGB-camera images.

3 Methods

Even though a wearable data glove, potentially integrated into an exoskeleton and its actuator system, might provide a valuable hand pose estimation [24, 6, 27], the information would not be context sensitive. A thorough computer vision approach may offer a rich overall scene analysis that includes the state of graspable objects, bare hands and of an exoskeleton. Further, the available space to mount sensors on an exoskeleton is rather limited.

For our computer vision based hand pose estimation we investigate a staged CNN approach [25]. This kind of 2D joint estimation has already been used in several other approaches of pose estimation [22, 12]. We train this CNN with synthetic data, similar to datasets like *SynthHands* [16] or *Rendered Handpose Dataset* (RHD) [28]. However, since no data generator is available that enables

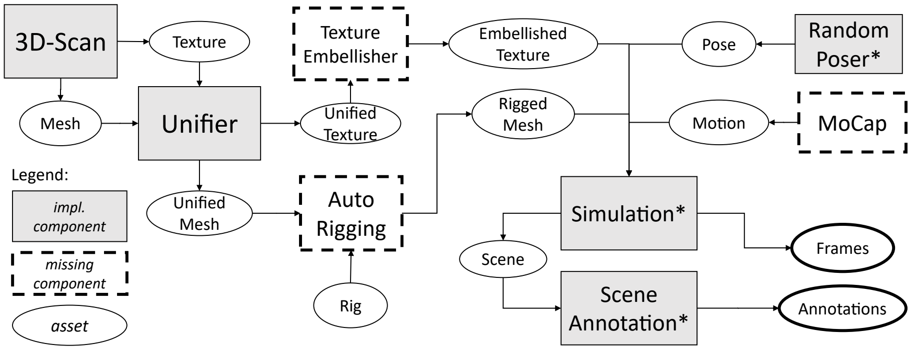

Fig. 3. Composition diagram of synthetic data generator. Components signified by * are implemented in *Blender*.

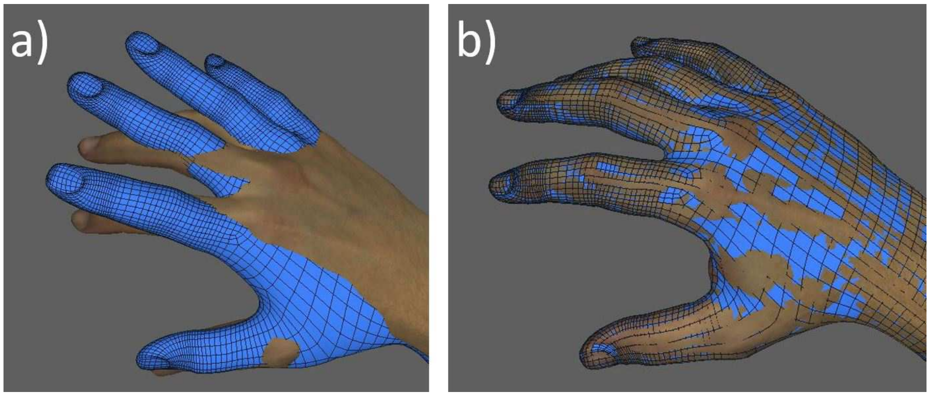

Fig. 4. Generalization of scanned meshes. (a) Base mesh (*blue*) aligned to the scanned mesh. (b) Base mesh wrapped around the scanned mesh.

to integrate an exoskeleton or to test different camera angles, we created our own data generator.

3.1 Synthetic Data Generator

Fig. 3 shows the composition of our data generator. First a 3D scan of a real hand is made with the *Artec 3D Eva*[1], a hand-held 3D scanner based on structured light. The scanner produces mesh and texture at the same time. Then *Wrap3.3*[2] is used as a `Unifier` to generalize the scanned mesh to a well structured and rigged mesh. *Wrap3.3* wraps a well-structured base mesh around an unstructured scanned mesh (Fig. 4), as such the output is a new generalized

[1] https://www.artec3d.com/de/portable-3D-scanners/artec-eva
[2] http://www.russian3dscanner.com/

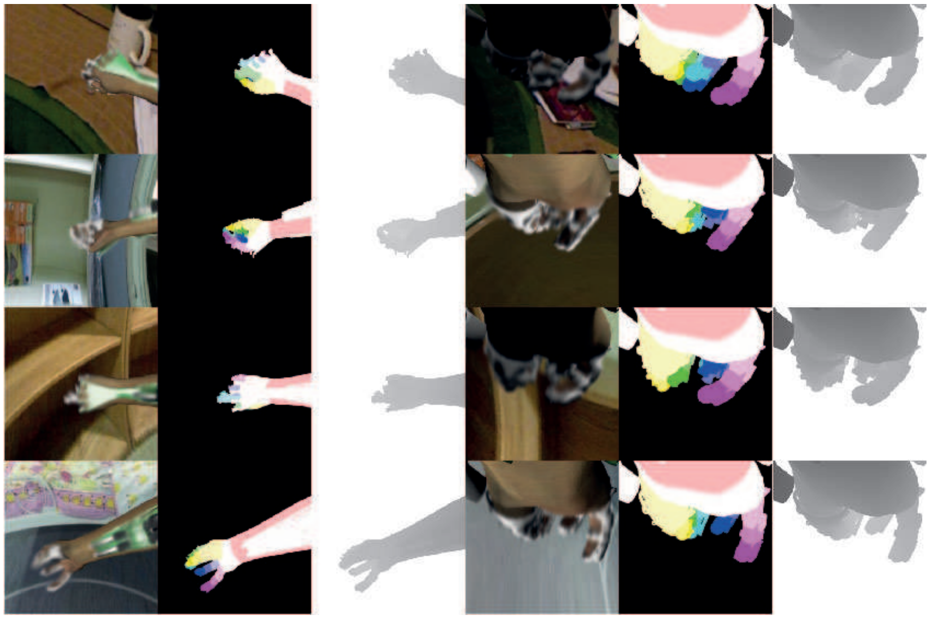

Fig. 5. Examples of our generated dataset. RGB-Maps of head-mounted camera (1st column), Segmentation-Map (column 2) and Depth-Map (column 3). Corresponding data for the wrist-mounted camera (columns 4-6). Details of the exoskeleton are blurred due to proprietary reasons at the time of submission.

mesh, easier to use for further steps, i.e. rigging and texturing. `AutoRigging` and `TextureEmbelishment` are not implemented yet. Editing textures and meshes is a laborious manual process. Therefore only one mesh and texture set is used up to now. For the `Simulation`, the open source 3D creation suite *Blender*[3] is used similar to [28, 18]. An early version of a CAD prototype of the exoskeleton is imported and aligned to the hand mesh in the `Simulation`. Finally two *Blender* add-ons are scripted (`RandomPoser` and `SceneAnnotation`). A main script loads and triggers these add-ons. It also manages the render process and randomizes the background, camera positions and light conditions.

The add-on `RandomPoser` performs pseudo random poses by applying random transformations along the kinematic chain. Our hand model has 17 bones (Fig. 6). The Degrees of Freedom (DoF) of the hand model have been put under virtual constraints to simulate articulated hand configurations by considering natural constraints and limits of collisions and rotations, as suggested by *DhaibaHand* and inspired by Jorg *et al.*.

The add-on `SceneAnnotation` stores all relevant information about the rendered scene in an XML-file. With our data generator we are able to produce

[3] https://www.blender.org/

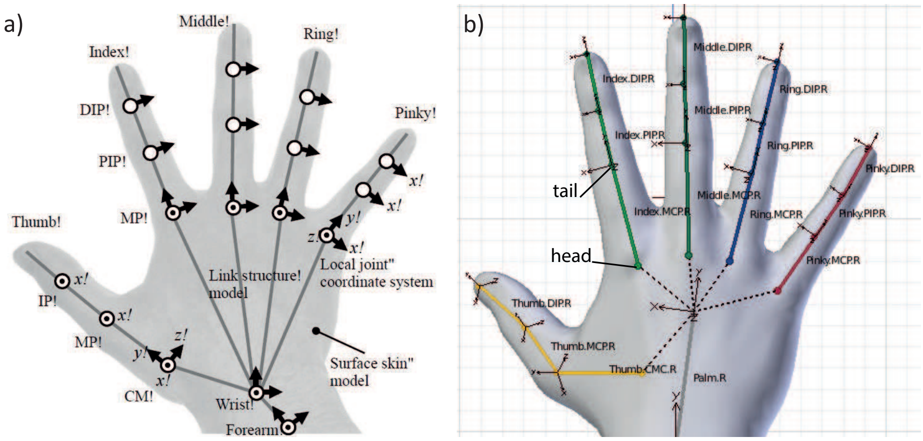

Fig. 6. Hand model of the simulation. (a) The DoF of *DhaibaHand* [7]. (b) Our hand model in *Blender*, each bone with head and tail vertex.

vast amounts of data automatically. Up to this point the following data has been extracted from a head- and wrist-mounted camera setup:

- **RGB-Map:** The reference image of visible light, with a resolution of 512∗512 and 24-bit PNG encoding. For background the indoor images of NYU Depth V1 dataset are used [21].
- **Depth-Map:** An 8-bit PNG gray scaled image of depth information. For the head camera viewpoint the first 100cm are linearly quantized and for the wrist viewpoint the first 50cm are quantized.
- **Segmentation-Map:** A 24-bit PNG encoded image where each part has its own color code.
- **Annotation:** An XML-file with coordinates of each component in world space (as $4*4$ *matrix*), in camera space and relative to the image plane (as *vertex*).

Each bone of our hand model is defined via two *vertices* (head and tail), 21 of them are used as keypoints. A *vertex* is a tuple represented as

$$vertex : (name, x, y, z, r, c, u, v, d), \tag{1}$$

which is parameterized as follows:

- *name* denotes if the *vertex* is head or tail of the bone.
- x, y, z are the relative coordinates in camera space with origin at the center of the camera.
- r, c are the pixel-based 2D coordinates on the image plane $(row, column)$ with the origin being the upper-left corner.
- u, v are the normalized 2D coordinates on the image plane with the origin being the lower-left corner.

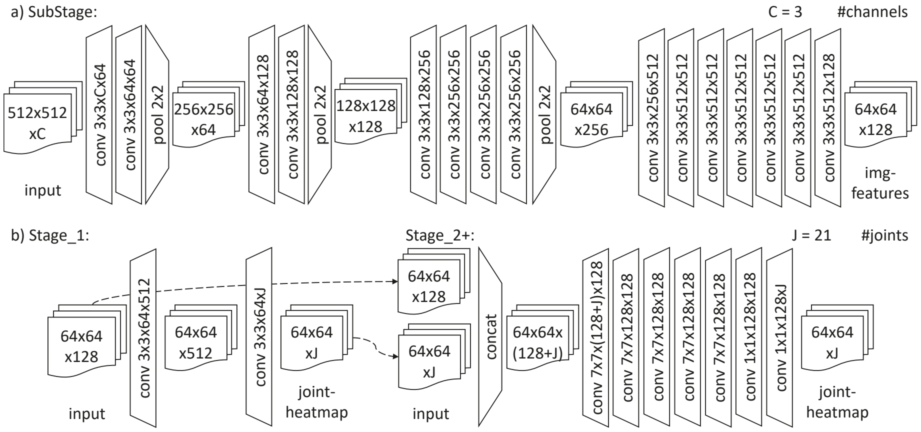

Fig. 7. Hand Pose Estimation with staged approach. (a) Shows the layers of SubStage for image feature extraction. (b) Shows the staged approach of [25], whereby img-features and joint-heatmap get concatenated for the subsequent stage.

– d is the depth value relative to the image plane.

We further used the random image adjustment functions of TensorFlow to randomize brightness, contrast, saturation and hue before training.

3.2 Pose Estimation Model

For the pose estimation a *TensorFlow* [1] version of Wei's *et al.* [25] CNN approach is used, adapted by Ho *et al.* [11]. The input layer is increased to $512{*}512{*}3$, since we used a GeForce GTX 1080 Ti with enough storage. Our hand model has 21 joints (including 5 fingertips), therefore the output is a heatmap of 21 channels of size $64 * 64$. Our model has one `SubStage` for extracting image features and six stages for pose estimation as recommended by Wei *et al.* [25]. Except for the first stage, each stage uses the concatenated image features and the heatmap of the previous stage as input (Fig. 7). As such, the output of each heatmap gets refined in a feed-forward approach. This enables the machine to learn filters to verify the spatial context between the estimated keypoints provided in the heatmap of the previous stage.

4 Experiment

The data generator is used to create 4,000 samples with bare hands and 4,000 samples with an exoskeleton. Both contain images of head- and wrist-mounted cameras (Fig. 5). For each of the $i : (1 \dots 21)$ joints a Gaussian map is created pixel-wise (m, n) with a radius of $r = 3$ at the joint position (x, y) stacked to a

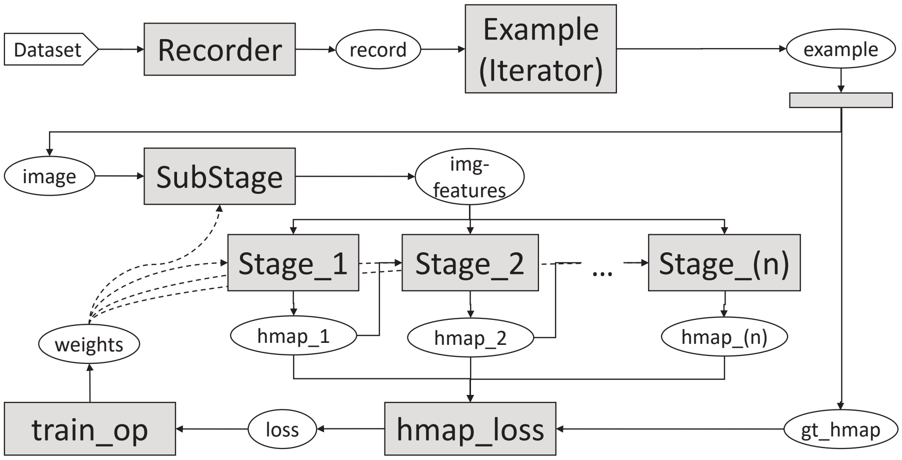

Fig. 8. Composite diagram of our CNN for hand pose estimation, inspired by Wei et al. [25], adapted by Ho *et al.* [11].

Table 1. Training 5 different models to compare head vs. wrist mounted cameras and bare hand vs. exoskeleton.

Model	Domain	Examples	Epochs	Steps	Time
head	HeadCam	2,000	20	40,000	3.3h
right	RightCam	2,000	20	40,000	3.3h
bare	HeadCam+RightCam	4,000	10	40,000	3.3h
exo	Exoskeleton (Head+Right)	4,000	10	40,000	3.3h
hybrid	bare+exo	8,000	10	80,000	6.6h

ground truth heatmap G of 21 $64 * 64$ channels and stored into a TensorFlow-Record together with the RGB reference image:

$$G_{mni} = exp\left(-\left((m-x_i)^2 + (n-y_i)^2\right)\frac{1}{2r^2}\right) \tag{2}$$

For training (Fig. 8) the total difference between the estimated heatmaps H of all $K = 6$ stages and the ground truth heatmap G is used as a cost function:

$$hmap\_loss(H,G) = \sum_{k=1}^{K} |H_k - G| \tag{3}$$

For evaluating the two camera setups and the domain of the hand with and without the exoskeleton, five models have been trained:

- `head`: Trained on images of bare hands observed from head-mounted camera
- `right`: Trained on images of bare hands observed from right wrist-mounted camera

- bare: Trained on images of head + right
- exo: Trained on images of hands with exoskeleton from head- and wrist-mounted camera
- hybrid: Trained on images of bare hands and hands with exoskeleton (bare + exo).

Each of these models are trained for about 40k steps, apart from hybrid, which is trained for 80k steps. The training of 40k steps takes just 3.3 hours performed by a GeForce GTX 1080 Ti (Table 1). These five models are cross-evaluated with four corresponding evaluation sets:

- head with 100 examples of head-mounted camera
- right with 100 examples of wrist-mounted camera
- bare with 200 examples of head + right
- exo with 200 examples of hands with exoskeleton.

For evaluation, the Euclidean distance between the estimated pose p and the ground truth pose g is used, whereby $K = 21$ denotes the number of keypoints each pose contains:

$$D(p, g) = \sum_{k=1}^{K} \|p_k - g_k\|_2 \tag{4}$$

Based on this error measure we test the performance of pose estimation on images of a head- versus a wrist-mounted camera and bare hand versus hands with exoskeleton. For comparison, the Percentage Correct Keypoints (PCK) [26] metric is used as in [25, 12, 22]. Though we do not normalize the PCK to the hand size, we normalize to the image aspect instead. So we do not give an extra penalty on small hands with lower resolution, just because they are more far from the image plane. Otherwise the evaluation set head would not be competitive with the evaluation set right. Since the image plane always has an aspect of $512 * 512$ pixels, an inaccuracy of 0.1 represents an error of 51.2 pixels.

5 Results

The y-axis of the PCK graphs (Fig. 9 and Fig. 10) show the percentage of examples that get estimated to an accepted accuracy, given as an increasing threshold on the x-axis. The accuracy is the total Euclidean distance of all keypoints normalized to the image aspect. Each line represents a model applied on an evaluation set. The earlier the line rises up to 100% the better the model performs on the evaluation set.

In Fig. 9, the PCK graphs of head- versus wrist-mounted cameras are shown. For qualitative examples, compare Fig. 11 (a) with (b). The evaluations head-on-right and right-on-head have low accuracy, because the model head has never seen examples of the domain right before and vice versa. Both these evaluations act as control groups to prove that head and right are actually in a separate

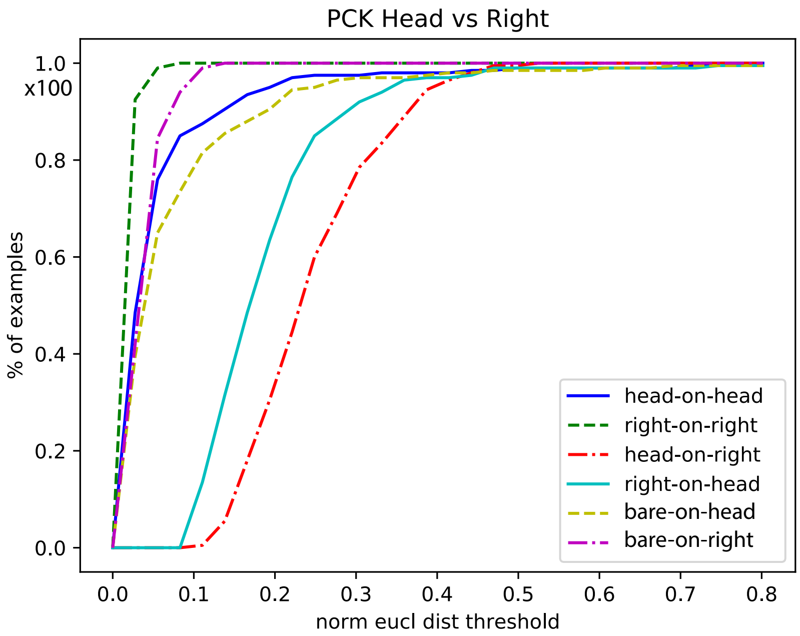

Fig. 9. Evaluation of Head versus Right. Each line describes a trained model applied on an evaluation set. *head-on-head*: A model trained and applied on head-mounted camera images. *right-on-right*: A model trained and applied on wrist-mounted camera images of the right hand. *head-on-right*: A model trained on head but applied on right wrist. *right-on-head*: Vice versa of *head-on-right*. *bare-on-head*: A model trained on both setups, but applied to head only. *bare-on-right*: A model trained on both setups, but applied to the right wrist.

domain. A noteworthy observation is that `right-on-right` perform much better than `head-on-head`. This leads us to the point that hand pose estimation with our CNN performs better on wrist-mounted cameras than on head-mounted cameras. The model `bare` shows a clear tradeoff between both domains.

In Fig. 10, the PCK graphs of bare hands versus hands with exoskeleton are shown. For qualitative examples, compare Fig. 11 (a,b) with (c,d). The control groups `exo-on-bare` and `bare-on-exo` perform relatively poorly. Therefore the appearance of the exoskeleton dominates the bare hand domain. The observation in `exo-on-exo` is that the appearance of the exoskeleton condition leads to a slightly better pose estimation than bare hands do. We assume that the visual features of the exoskeleton are more significant, easier to detect and thus enhance the recognizability of the hand. Furthermore, the model `hybrid` also performs well on `exo`, with just a small reduction on `bare`. With this we conclude that a `hybrid` model with an extended training session may work well in the overall

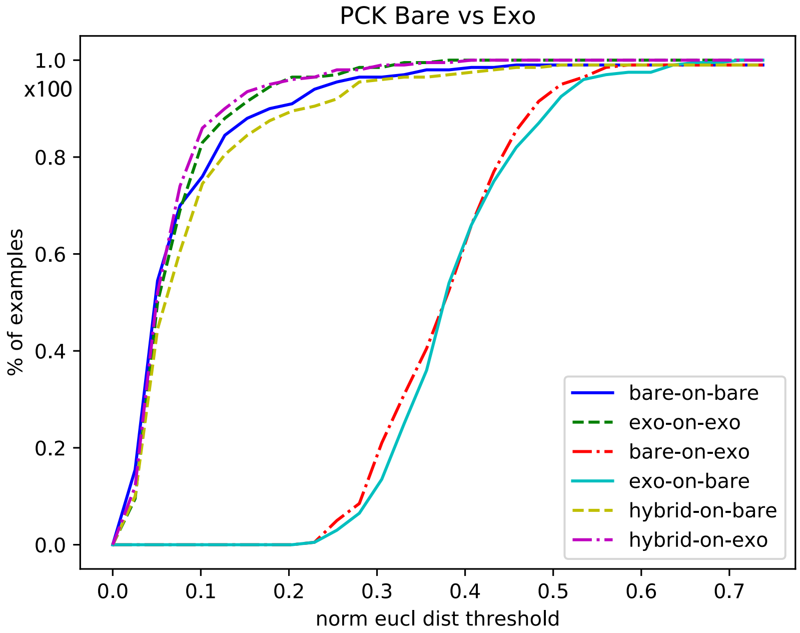

Fig. 10. Evaluation of Bare versus Exo. Each line describes a trained model applied on an evaluation set. *bare-on-bare*: A model trained and applied on images of bare hands. *exo-on-exo*: A model trained and applied on images of hands with exoskeleton. *bare-on-exo*: A model trained on bare hands, but applied to exoskeleton images. *exo-on-bare* vice versa of *bare-on-exo*. *hybrid-on-bare*: A model trained on both domains, but applied on bare hands only. *hybrid-on-exo*: A model trained on both domains, but applied to exoskeleton images only.

project application. In summary, we are able to estimate poses of the exoskeleton and of bare hands from head-mounted and wrist-mounted cameras within the context of our simulation, whereas the wrist-mounted camera performs better with respect to the metric presented above.

6 Discussion

Although the occlusion of the fingers on wrist-mounted camera images might be more severe, it does not seem to drastically affect the pose estimation. Interestingly, Chan et al. [4] have already shown that for the task of grasp classification from a wrist-mounted camera, the performance is slightly better than from a head-mounted camera. In our evaluation we observed that the hand pose estimation performs significantly better on wrist-mounted cameras.

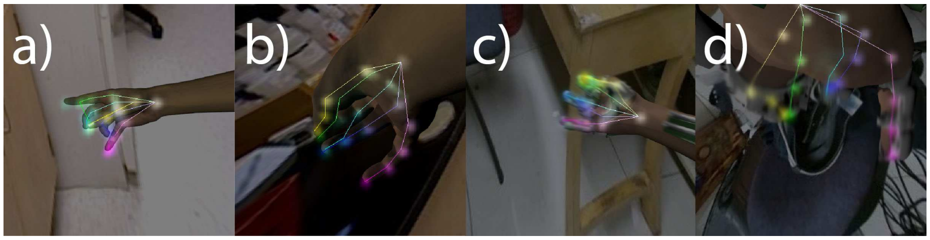

Fig. 11. Qualitative examples of pose estimation. The dots are from the heatmap as estimated, the lines connect the maxima of the heatmap by given topology: (a) Head-mounted camera on bare hand, (b) wrist-mounted camera on bare hand, (c) head-mounted camera on exoskeleton and (d) wrist-mounted camera on exoskeleton. Details of the exoskeleton are blurred due to proprietary reasons at the time of submission.

In addition to the technical challenges of hand pose estimation from an egocentric viewpoint we discuss some subjective issues of the camera setups. During lab testing and on qualitative examples from other datasets we developed an impression of the usability and some of its attributes. We extend the discussion here with insights from chest-mounted camera setups. Chest-mounted cameras are used in *EgoDexter* [16], *UCI-EGO* [20] and *GUN71* [19]. Head- or chest-mounted cameras have an aesthetic issue, because the user could feel uncomfortable with the camera at these positions. To wear a camera on the head might be more uncomfortable than on the chest. If miniaturized, chest cameras could be integrated into textiles. The dataset *EgoHands* [2] qualitatively demonstrates the shortcoming of head-mounted cameras. Obviously the user has to put the hands into the Field of View (FoV) towards a grasp goal during a potential application, such that all relevant scene elements can be observed from the camera. However, we recognize in datasets like *EgoHands* and in a self test that the hand often disappears beyond the bottom image edge. Usually the user mainly fixates with his eyes and avoids to move his whole head. We assume that this behavior might be intensified by wearing a sensor on the head. Thus a user would need to get used to the setup and needs to learn to look straight and actively keep track of his interaction zone. Nevertheless, it might be more intuitive than a wrist- or chest-mounted camera. Yet, in such setups the user has less control of the camera's FoV. As long as a display of the camera is not present, the user will not be able to relate to what the camera might see or not. In the case of the wrist-mounted camera setup it is ensured that it will fixate the wearing hand, but the graspable object and the other hand might not always be in the FoV. The dataset *EgoDexter* [16] contains many examples in which the hand is beyond the FoV of the camera. Nevertheless, the workspace of the chest-mounted camera is much easier to analyze because the observation space is relatively stationary in front of the user, whereas the head- or wrist-mounted camera could move quite freely.

Before we can make further design decisions the different camera setups should be evaluated with subjects. Especially the user acceptance of aesthetic and comfort should be investigated. As a result of this paper the wrist-mounted camera setup has promising properties and should be tested in further scenarios of the wearable robotic grasp assistance application.

7 Conclusions

In this paper we presented a customizable data generator with *Blender* and used the approach of a staged CNN to evaluate the performance of hand pose estimation for egocentric viewpoints. The data generator is able to include a wearable exoskeleton making it unique in the field of hand simulation frameworks. We observed in our simulation that hand pose estimation on wrist-mounted camera images performs significantly better than on head-mounted camera images in terms of PCK scores. Furthermore we observe that our exoskeleton defines a new domain of appearance by covering large portions of the assisted hand. Remarkably, the selected CNN approach seems to be powerful enough to learn and handle all domains captured from head- and wrist-mounted cameras with exoskeleton and bare hands in one hybrid model.

In the future we plan to classify all joints with fully-connected layers to a set of grasp poses and compare this approach with other grasp pose classifiers. We plan to extend our data generator in several aspects based on further assets, i.e. including more meshes, more textures, improved shaders and full 3D scenes instead of only 2D backgrounds. Furthermore, we want to look into the transfer of synthetic simulations to real data, e.g. with a Generative Adversarial Network (GAN) [9] as pursued for example in [15]. Instead of enhancing synthetic data to a natural look, it might be easier to downgrade real data to a synthetic look, or rather a generalization in between these conditions could be feasible.

However, we will also generate real data. We currently develop a setup for tracking hands in a motion capture laboratory with a marker-based system and synchronized sensors. In further studies, the influence of a real exoskeleton prototype will be investigated in comparison to simulated prototypes.

Acknowledgements

We gratefully acknowledge the Baden-Württemberg Foundation for supporting KONSENS-NHE (NEU007/3) in the program neurorobotics as well as the Federal Ministry of Education and Research (BMBF) for funding the projects MoCap 4.0 (03FH015IN6) and KollRo 4.0 (13FH049PX5). The exoskeleton CAD model prototype was kindly provided by Jonathan Eckstein from the University of Stuttgart.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S.,

Murray, D.G., Steiner, B., Tucker, P.A., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zhang, X.: Tensorflow: A system for large-scale machine learning. CoRR **abs/1605.08695** (2016), http://arxiv.org/abs/1605.08695

2. Bambach, S., Lee, S., Crandall, D.J., Yu, C.: Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
3. Barsoum, E.: Articulated hand pose estimation review. CoRR **abs/1604.06195** (2016), http://arxiv.org/abs/1604.06195
4. Chan, C., Chen, S., Xie, P., Chang, C., Sun, M.: Recognition from hand cameras. CoRR **abs/1512.01881** (2015), http://arxiv.org/abs/1512.01881
5. Chen, X.: Awesome work on hand pose estimation. GitHub (2018), https://github.com/xinghaochen/awesome-hand-pose-estimation
6. Dipietro, L., Sabatini, A.M., Dario, P.: A survey of glove-based systems and their applications. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **38**(4), 461–482 (July 2008). https://doi.org/10.1109/TSMCC.2008.923862
7. Endo, Y., Tada, M., Mochimaru, M.: Reconstructing individual hand models from motion capture data. Journal of Computational Design and Engineering **1**(1), 1 – 12 (2014). https://doi.org/https://doi.org/10.7315/JCDE.2014.001, http://www.sciencedirect.com/science/article/pii/S2288430014500012
8. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding **108**(1), 52 – 73 (2007). https://doi.org/https://doi.org/10.1016/j.cviu.2006.10.012, http://www.sciencedirect.com/science/article/pii/S1077314206002281, special Issue on Vision for Human-Computer Interaction
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
11. Ho, T.: Convolutional pose machines - tensorflow. GitHub (2018), https://github.com/timctho/convolutional-pose-machines-tensorflow
12. Iqbal, U., Molchanov, P., Breuel, T., Gall, J., Kautz, J.: Hand pose estimation via latent 2.5d heatmap regression. CoRR **abs/1804.09534** (2018), http://arxiv.org/abs/1804.09534
13. Kim, D., Hilliges, O., Izadi, S., Butler, A.D., Chen, J., Oikonomidis, I., Olivier, P.: Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology. pp. 167–176. UIST '12, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2380116.2380139, http://doi.acm.org/10.1145/2380116.2380139
14. Maekawa, T., Yanagisawa, Y., Kishino, Y., Ishiguro, K., Kamei, K., Sakurai, Y., Okadome, T.: Object-based activity recognition with heterogeneous sensors on wrist. In: Floréen, P., Krüger, A., Spasojevic, M. (eds.) Pervasive Computing. pp. 246–264. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
15. Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., Theobalt, C.: Ganerated hands for real-time 3d hand tracking from monocular RGB. CoRR **abs/1712.01057** (2017), http://arxiv.org/abs/1712.01057

16. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In: Proceedings of International Conference on Computer Vision (ICCV) (October 2017), http://handtracker.mpi-inf.mpg.de/projects/OccludedHands/
17. Pisharady, P., Saerbeck, M.: Recent methods and databases in vision-based hand gesture recognition: A review **141**, 152–165 (12 2015)
18. Rajpura, P.S., Hegde, R.S., Bojinov, H.: Object detection using deep cnns trained on synthetic images. CoRR **abs/1706.06782** (2017), http://arxiv.org/abs/1706.06782
19. Rogez, G., Supancic, J.S., Ramanan, D.: Understanding everyday hands in action from rgb-d images. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3889–3897 (Dec 2015). https://doi.org/10.1109/ICCV.2015.443
20. Rogez, G., III, J.S.S., Khademi, M., Montiel, J.M.M., Ramanan, D.: 3d hand pose detection in egocentric RGB-D images. CoRR **abs/1412.0065** (2014), http://arxiv.org/abs/1412.0065
21. Silberman, N., Fergus, R.: Indoor scene segmentation using a structured light sensor. In: Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition (2011)
22. Simon, T., Joo, H., Matthews, I.A., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. CoRR **abs/1704.07809** (2017), http://arxiv.org/abs/1704.07809
23. Soekadar, S.R., Witkowski, M., Gómez, C., Opisso, E., Medina, J., Cortese, M., Cempini, M., Carrozza, M.C., Cohen, L.G., Birbaumer, N., Vitiello, N.: Hybrid eeg/eog-based brain/neural hand exoskeleton restores fully independent daily living activities after quadriplegia. Science Robotics **1**(1) (2016). https://doi.org/10.1126/scirobotics.aag3296, http://robotics.sciencemag.org/content/1/1/eaag3296
24. Sturman, D.J., Zeltzer, D.: A survey of glove-based input. IEEE Computer Graphics and Applications **14**(1), 30–39 (Jan 1994). https://doi.org/10.1109/38.250916
25. Wei, S., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. CoRR **abs/1602.00134** (2016), http://arxiv.org/abs/1602.00134
26. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(12), 2878–2890 (Dec 2013). https://doi.org/10.1109/TPAMI.2012.261
27. Zhang, X., Chen, X., Li, Y., Lantz, V., Wang, K., Yang, J.: A framework for hand gesture recognition based on accelerometer and emg sensors. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans **41**(6), 1064–1076 (Nov 2011). https://doi.org/10.1109/TSMCA.2011.2116004
28. Zimmermann, C., Brox, T.: Learning to estimate 3d hand pose from single RGB images. CoRR **abs/1705.01389** (2017), http://arxiv.org/abs/1705.01389