Back To RGB: Deep Articulated Hand Pose Estimation From a Single Camera Image

Wan-Duo Kurt Ma, J.P. Lewis, Marcus Frean and David Balduzzi
School of Engineering and Computer Science,
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand

***Abstract*—In this work, we demonstrate a method called Deep Hand Pose Machine(DHPM) that effectively detects the anatomical joints in the human hand based on single RGB images. Current state-of-the-art methods are able to robustly infer hand poses from RGB-D images. However, the depth map from an infrared camera does not operate well under direct sunlight. Performing hand tracking outdoors using depth sensors results in unreliable depth information and inaccurate poses. For this reason we were motivated to create this method which solely utilizes ordinary RGB image without additional depth information. Our approach adapts the pose machine algorithm, which has been used in the past to detect human body joints. We perform pose machine training on synthetic data to accurately predict the position of the joints in a real hand image.**

I. Introduction

Hand pose plays a crucial role in manipulating objects and delivering messages via sign language, interactions with a computer, and mobile devices. Hand pose estimation is the task of automatically estimating the hand pose in order to support fluid human-computer interaction. Unfortunately, the motion blur from an optical system, object occlusion, and partial visibility can complicate this task. To tackle these problems, there are a few commercial products that have been developed such as *Leap Motion*[1] which utilizes IR cameras to capture the depth information of the hand and to track the skeletal joints of the user. New approaches to hand tracking are still an active field for researchers.

When creating a physical simulation, one could treat hand structure as a kinematic skeletal model. During runtime solving, the optimization process could involve kinematic constraints for more accurate refinements. Additional anatomical and physical information, such as body fat and inertia, is usually unnecessary when concentrating solely on solving the problems associated with hand pose estimation.

Since the success of deep learning, which has helped to master the challenges presented by ImageNet [1], the number of researchers utilizing deep learning in their different fields of expertise has greatly increased. A deep learning model is compromised of a large number of concatenated layers, which are parameterized by a large number of weights. However, collecting the necessary training data often relies on extensive manual effort.

[1]https://www.leapmotion.com/

978-1-5386-4276-4/17/$31.00 ©2017 IEEE

We introduce Deep Hand Pose Machine (DHPM) and Synthetic Hand training data generation in this paper. The pose machine trained on synthetic data is based on the similarities between virtual and real hand models. They both share information such as anatomical structure and texture. Additionally, a deep learning model needs a huge amount of data to make learning tractable. Based on these factors, we show (in section III) our work on hand modeling and data generation.

DHPM takes utilizes **pose machine** [2] [3] to provide a way to sequentially localize the joints in the human body by considering both image and spatial context directly. The hand involves both hierarchical articulated motion as well as the possibility of occlusion. To address this, we instead use a level-wise grouping approach for the design of DHPM.

Our main contributions are (a) synthetic hand data generation, including a method of animating the hand model, and camera setups, (b) the construction of an effective sequential learning model that considers the spatial context between joints; and (c) data augmentation on YCbCr color space that avoids the color mismatch inherent in a virtual hand model. We also provide an analysis of both the validation dataset and segmented real hand images.

II. Related Work

Hand tracking and pose estimation has been the subject of research for several decades. Since the release of the commercial RGB-D camera Microsoft Kinect and its development api, attention has substantially shifted from RGB cameras to RGB-D cameras. For instance, [4] [5], using multiple RGB-D cameras for hand capturing to avoid self-occlusions. On the other hand, single depth camera approaches are also popular under restricted ranges of capturing [6][7][8]. These non-temporally coherent methods sometimes suffer from unreliable depth information that makes the optimization converge inaccurately.

Synthetic data acquisition from a virtual model is also a promising solution for simulating hand pose. Most researchers, for example [9][10][11][12], who have been rendering depth images for training have utilized a fixed camera and this makes capturing the entire possible range of a hand motion problematic. Data generation from an egocentric point of view [13] provides a way to improve the prediction for portable devices such as Google Glasses or a Go-Pro camera. Their

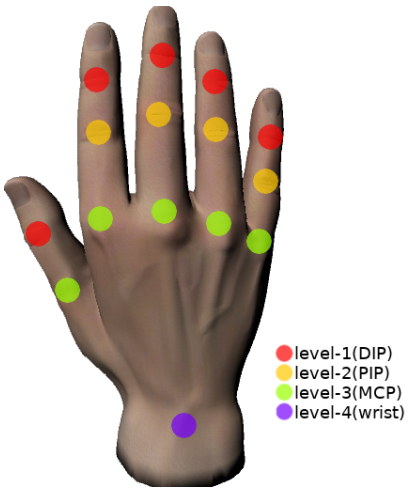

Fig. 1. Our articulated hand model with different marks representing the level of the hand and the phalanges being; level-1, the distal interphalangeal joints (DIP); level-2, the proximal interphalangeal joints (PIP); level-3, metacarpophalangeal joints (MCP) and level-4: wrist. In this paper, we only address DIP, MCP, and the wrist.

method also requires more user effort, as they must refine 3d pose estimation from 2d annotations, since it has only semi-automatic labeling.

Pose estimation techniques can be categorized into discriminative and generative methods. Generative methods, as seen in [14] [15] [16], fit a template model onto raw data, such as the depth information we mentioned previously. The optimization process is performed at runtime by minimizing the target function with some kinematic constraints. Simultaneously, the temporal context is also considered in order to avoid jittering.

In contrast, discriminative methods [17] [8] [18] estimate the pose through an off-line pre-trained model. Usually solving performance is more efficient when compared to the iteratively optimized generative approach. The pre-trained model might lack a full-range of training data thus making the prediction made by the discriminative approach unreliable.

III. Synthetic Data

The motivation behind training with synthetic data is because of the similarity between virtual data and real data such as: contours, textures in vision domain, and the same number of phalanx bones anatomically speaking. Instead of spending much more time on collecting training data and labeling, we demonstrate that synthetic images rendered from a 3d object are able to be used to construct our training dataset.

A. Hand model

Fig. 1 shows our articulated hand model with textures, marked on each joint to denote the level of the fingers. Level-1 to level-4 the distal interphalangeal joints(DIP), proximal interphalangeal joints(PIP), metacarpophalangeal joints(MCP), and the wrist are represented. There are in total 20 degrees of freedoms [2] of movement in our hand model that we animated into

[2]each DIP and PIP has 1 degree of freedom, MCP has 2 degrees of freedom

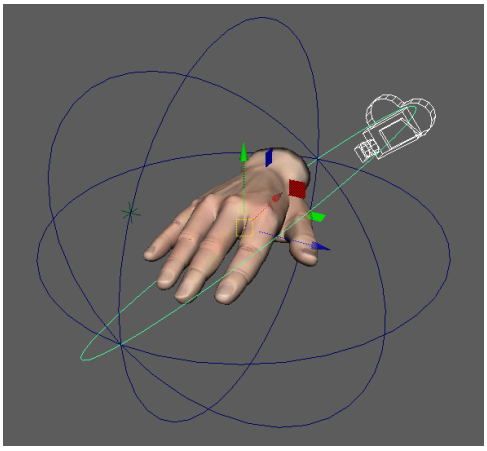

Fig. 2. Generation of training data using the commercial software *Autodesk Maya*. Given a hand model, we create four trajectories centered on the middle MCP with a camera attached. Each trajectory will be evenly partitioned into six positions, resulting in each hand pose. At frame i: $\left(x_p^c\right)_i$ has 24 training rendered images.

Fig. 3. An example showing the sequence of animation under category $c \leftarrow [1,1,1,1,1]$, which represents that all fingers are able to crunch simultaneously with four different camera views listed in rows. The camera configurations such as focal length and field of views (FOV) are manually adjusted, to ensure the whole hand is visible in the image.

a specific category c where $c \in \{00001, 00010, ..., 11111\}$. Fig. 3 shows binary coded configuration of the hand's finger movements. Each category is a sequence of hand movements from the neutral pose (with a stretched palm) to the target pose. For instance $c \leftarrow 00001$ is from neutral to the pose where the little finger is fully crunched, $c \leftarrow 11111$ means the sequence from neutral hand to fist. In total, there are 31 categories in our settings.

B. Rendering

In order to capture all possible hand footage $x \in \mathbb{R}^{w \times h}$, we created four trajectories around the hand model with a camera attached. See Fig. 2 for more detail. Each trajectory is evenly partitioned into four positions that allows the camera to render the hand model. Under the same configuration of lighting and camera parameters, we sampled a training element by rendering the image, as well as its 2d location joint labels.

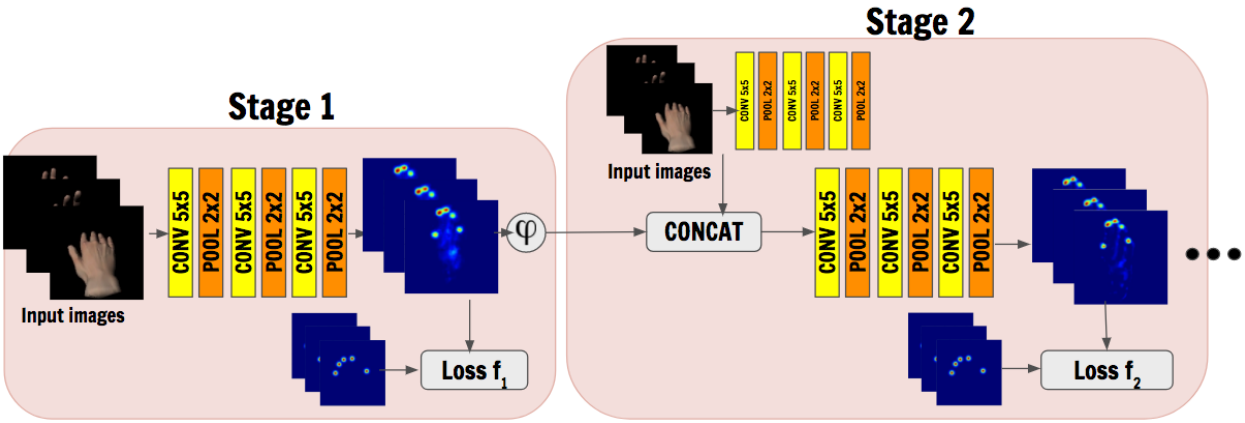

Fig. 4. **Deep hand pose machine (DHPM)** is a deep architecture comprising T stages, where each stage is implemented using a convolutional neuron network. The beginning stage S_1 only utilizes the incoming training images. Similarly, the subsequent stages S_t where $t \in \{2, ..., T\}$ takes the same images as well as the feature ϕ produced from previous stage S_{t-1}. ϕ is the localization feature function that is applied on the result of stage S_{t-1}. This architecture effectively captures long-range spatial information [2].

IV. Method

A. Deep Hand Pose Machine

Instead of detecting the human hand joints individually, we perform a level-wise grouping based strategy that makes the training tractable. We denote the location of p-th landmark $Y_p \in \chi \subset \mathbb{R}^2$ where $p \in [1, ..., P]$ and χ is a set of all image locations. Training belief label $b_* \in \mathbb{R}^{w \times h}$ is then applied in the form of n Gaussian distributions around each joint $p \in P_l$ where $P_l \in \mathbb{R}^{n \times 2}$ is a set of n 2d joint locations at level l. For instance, $P_1 \in \mathbb{R}^{5 \times 2}$ represents a set of finger tip locations.

A pose machine model M_l is a network that is composed of several stages S_t where $t \in \{1, ..., T\}$ is the indexes of the stages. In the following we drop l for notational simplicity.

Each stage has its own predictor $g_t(\cdot)$ which is trained to predict the specific G. One could think of the result from the predictor as the score of a heat map, denoting the confidence of the predicted result. Our DHPM pipeline is shown in Fig 4.

The predictor g_1 at the first stage S_1 is quite different from the others since there is no previous information coming through. It takes the raw image x and produces the belief values.

$$g_1(x) \rightarrow b_1 \tag{1}$$

where $b_1 \in \mathbb{R}^{w \times h}$ is the activation from S_1. All stages S_t where $t \in \{2, ..., T\}$ except the first one will take the spatial context $\phi_{t>1}$ and the processed input data x':

$$g_t(x', \phi_t(z, b_{t-1})) \rightarrow b_t \tag{2}$$

where $\phi_{t>1}$ is the receptive field that is mentioned in [2], which can refine the result in the next stage, and x' is coming from the image data x with a few convolutional layers on top.

B. Convolutional neural network

In the previous section we showed how the architecture of *DHPM* is constructed. The parameters could be learned from any number of optimization approaches, such as random forest or convolutional neural network (CNN), involved in pose machine. Our network architecture is comprised of five blocks of layers and followed by two 1x1 fully connected convolution layers [19] . Each block contains a convolutional layer with various kernel sizes, the non-linear activation function $ReLU$, and a batch-normalization layer. Such a configuration has been proved to have the ability to learn a deep model effectively.

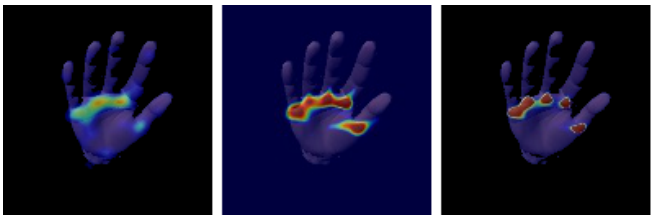

Fig. 5. The output heat map from each stage, overlapped with the original input source. From left to right shows the results of S_1, S_2 and S_3 respectively. We see that the distribution gradually converges on the locations of the MCP joints. In this case the blurry heat distribution gradually converges on MCP joints.

C. Data augmentation

To prevent creating a huge dataset off-line, we perform data augmentation during training. Specifically, the image is randomly rotated, scaled, and translated, by sampling the parameters r, s, and t from a uniform distribution with:

$$\begin{aligned} r &\sim U(0, 360) \\ s &\sim U(0.8, 1.2) \\ t_i &\sim U(-15, 15) \end{aligned} \tag{3}$$

where r is the angle of the R rotation matrix, s are the diagonal elements of the scaling matrix and t_i is the translation vector

Fig. 6. The result of YCbCr hand image perturbation in data augmentation. This approach allows the training to accommodate color mismatch from the synthetic dataset.

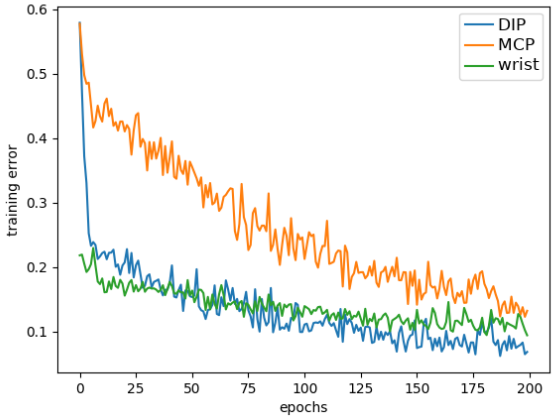

Fig. 7. The training error during 200 epochs with 3 training models on DIP, MCP, and the wrist respectively. This shows the degree of training difficulty from MCP, DIP to wrist.

according to the axis $i \in \{1, 2, 3\}$ and forming the vector t.

$$\begin{aligned} x^* &= R \cdot S \cdot x + t \\ y^* &= R \cdot S \cdot y + t \end{aligned} \tag{4}$$

Additionally, to eliminate the skin color bias in our training data, we also perform data augmentation in color space. In this work we manipulate the luminance and chrominance of a given image during training. Namely, each training datum x is formed in RGB color space. We then transform it into YCbCr color space with scholastic perturbation:

$$\begin{aligned} Y' &= Y + \epsilon_Y \\ Cb' &= Cb + \epsilon_{Cb} \\ Cr' &= Cr + \epsilon_{Cr} \end{aligned} \tag{5}$$

where ϵ_Y,ϵ_{Cb}, ϵ_{Cr} is the perturbation factors which samples from the Gaussian with unit variance and zero mean.

D. Learning algorithm

The problem of vanishing gradients [2] [20] where the magnitude of the learned gradient will decrease exponentially with n while training n-layers when back-propagated towards the front layer. To alleviate such problem, intermediate supervision is crucial that boosts learned gradients. There are many ways to solve this problem in the machine learning community, for instance skipping the connection model such in ResNet [21].

The convolutional pose machine proposed by [2] can be considered as an intermediate supervision methods that avoids a vanishing gradient [20]. More specifically, the convolutional pose machine is repeatedly applied as a loss function at each stage in order to boost the learning gradient in the deep model. The loss f_t at stage S_t minimizes the matrix norm of the objective:

$$f_t = ||b_t - b_*||_F^2 \tag{6}$$

By summing all the objectives we obtain:

$$F = \sum_{t=1}^{T} f_t \tag{7}$$

We used $Adam$ optimizer [22] for training with learning rate 0.0002, all lasso weight regularization 0.0001, dropout 0.001, batch size 16.

V. Evaluation

In this section we show results of our experiments in predicting a validation dataset (Fig. 8) and applying the assorted hand images from internet (Fig. 9). We apply our algorithms on the following three levels: Distal Interphalangeal (DIP), metacarpophalangeal (MCP) and wrist from left column to right respectively in each subject. Fig. 7 illustrates our training error in 200 epoches.

Fig. 8 shows experiments on the validation dataset, which are not available to DHPM during training. The three categories show how the belief heat-map increases confidence around the target as the number of stages get higher. Additionally, the invariance to the affine transformation of hand shows DHPM is generalized to learn structurally. For instance, it can recognize DIP from those similar features around finger tips.

Fig. 9 shows experiments on real hand images downloaded from the internet with background removal. Here we are interested in how DHPM which is trained on synthetic images generalizes to real images. The first block column shows the prediction based on DIP footage, with the correct exterior finger joints instead of the interior information. which is sensitive to the edge of the hand. The MCP experiment shows same effect (comparing Fig. 8) and the wrist shows reliable prediction.

VI. Conclusion

We demonstrated a technique for predicting the placement of the hand joints when the given a single RGB image.

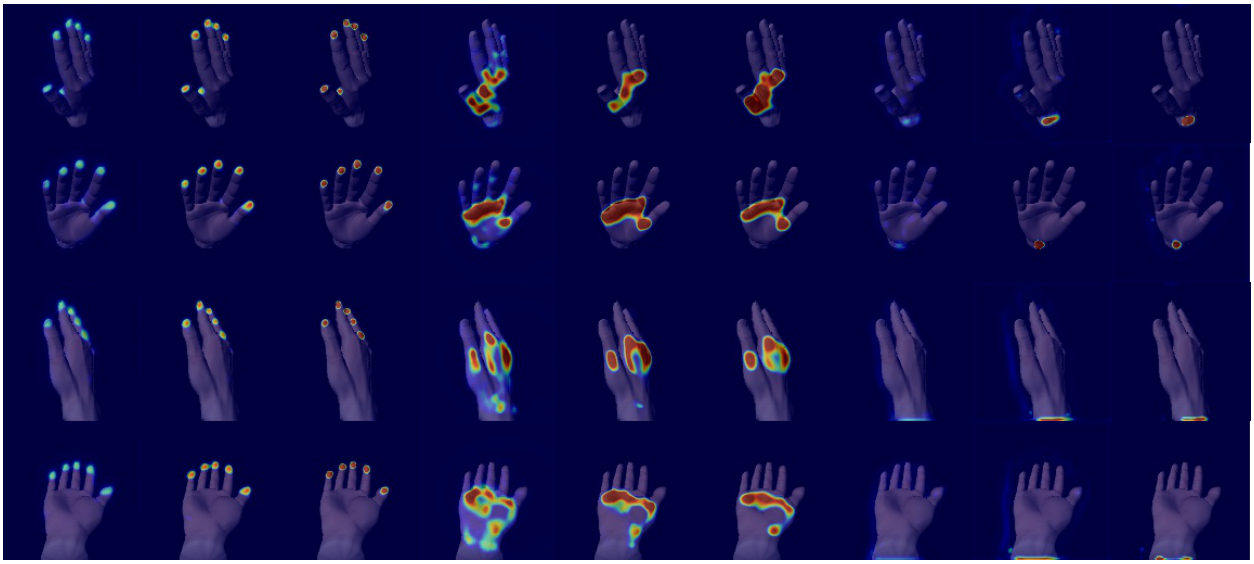

Fig. 8. Prediction of the validation dataset. Each row shows specific hand pose, with the prediction of DIP, MCP, and wrist in each block column. All three elements in such block represents the heat map b_t from S_t where $t \in \{1, 2, 3\}$ from left to right. The experiment shows how diffuse heat maps b_i become concentrated at subsequent stages b_j where $i < j$. For instance in second block column, the heatmap b_1 is scattered widely on whole hand image from MCP prediction b_1; and it gradually converges to MCP joints at stage 3, namely b_3.

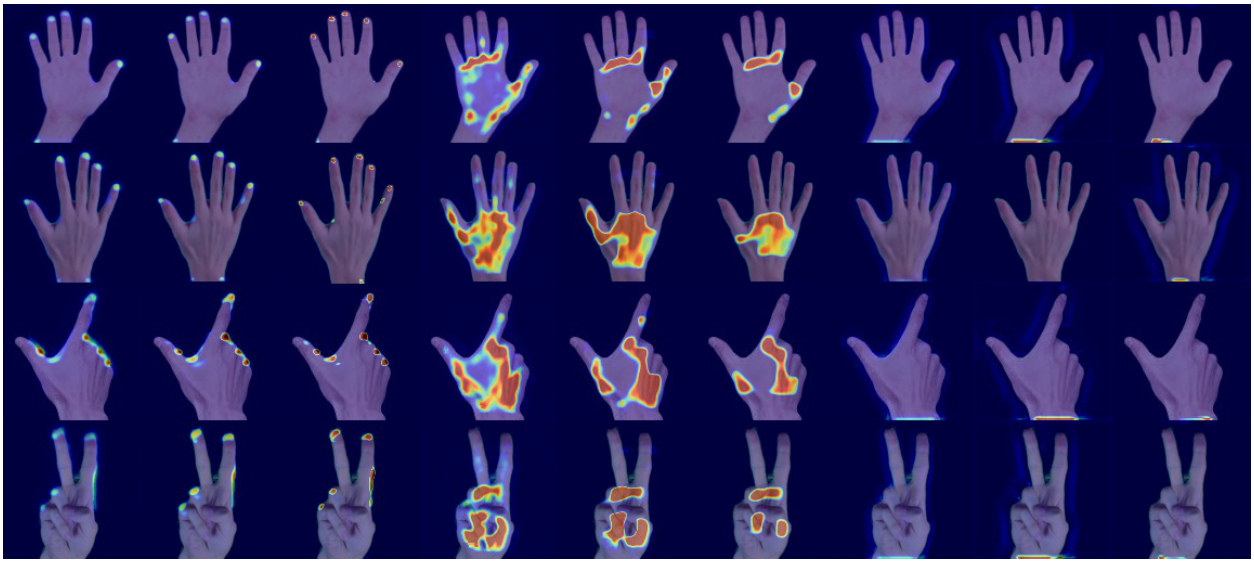

Fig. 9. Prediction of real hand images from the internet with background removal. The limitation of DHPM is in predicting hidden information, as can be seen in the third and fourth rows.

The Deep Hand Pose Machine(DHPM) uses receptive field for spatial context around the joint that improves the result. Additionally, we showed our level-wise learning for the complicated hand structure that makes the learned model tractable. In the future we would like to solve the domain adaptation problems for reducing the gap between training and test data.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[2] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[3] V. Ramakrishna, D. Munoz, M. Hebert, A. J. Bagnell, and Y. Sheikh, "Pose machines: Articulated pose estimation via inference machines," in *ECCV*, 2014.

[4] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 169:1–169:10, Sep. 2014.

[5] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive

markerless articulated hand motion tracking using RGB and depth data," Dec 2013.

[6] D. Tzionas and J. Gall, "A comparison of directional distances for hand pose estimation," pp. 131–141, 2013.

[7] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," pp. 1106–1113, June 2014.

[8] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," pp. 824–832, June 2015.

[9] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof, *A Framework for Articulated Hand Pose Estimation and Evaluation*. Springer International Publishing, 2015, pp. 41–52.

[10] G. Rogez, J. S. Supancic, and D. Ramanan, "Understanding everyday hands in action from RGB-D images," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3889–3897.

[11] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 3456–3462.

[12] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3d training data for fine hand pose estimation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 4957–4965.

[13] G. Rogez, M. Khademi, J. S. Supančič III, J. M. M. Montiel, and D. Ramanan, *3D Hand Pose Detection in Egocentric RGB-D Images*. Cham: Springer International Publishing, 2015, pp. 356–371.

[14] N. K. Iason Oikonomidis and A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2011, pp. 101.1–101.11.

[15] S. Melax, L. Keselman, and S. Orsten, "Dynamics based 3d skeletal hand tracking," 2017.

[16] A. Tagliasacchi, M. Schroeder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly, "Robust articulated-ICP for real-time hand tracking," *Computer Graphics Forum (Symposium on Geometry Processing)*, vol. 34, no. 5, 2015.

[17] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun, *Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests*. Springer Berlin Heidelberg, 2012, pp. 852–863.

[18] D. Tang, H. J. Chang, A. Tejani, and T. K. Kim, "Latent regression forest: Structured estimation of 3d articulated hand posture," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 3786–3793.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CVPR (to appear)*, Nov. 2015.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization." *CoRR*, vol. abs/1412.6980, 2014.