# Supplementary Material for the Paper "DeepHPS: End-to-end Estimation of 3D Hand Pose and Shape by Learning from Synthetic Depth"

Jameel Malik[1], Ahmed Elhayek[1], Fabrizio Nunnari[2], Kiran Varanasi[1],
Kiarash Tamaddon[2], Alexis Heloir[2], and Didier Stricker[1]

[1]*AV group, DFKI Kaiserslautern, Germany*
[2]*DFKI-MMCI, SLSI group, Saarbruecken, Germany*
{jameel.malik,ahmed.elhayek,fabrizio.nunnari,kiran.varanasi,
kiarash.tamaddon,alexis.heloir,didier.stricker}@dfki.de

## 1. The HPSL Gradient Derivation

In this section, we provide the detailed mathematical derivation of the **HPSL** functions. For backward-pass in the **HPSL**, we compute gradients of the following equation with respect to the layer inputs:

$$\textbf{HPSL}(\Theta, \beta, \alpha) = (\, \boldsymbol{F}(\Theta, \alpha)\,, \Upsilon(\Theta, \beta, \alpha)\,). \qquad (1)$$

Each vertex $\mathrm{v}_{\varkappa}$ in the reconstructed hand morphable model $\Psi$ is deformed using linear blend skinning [3]. Hence, for every vertex, the gradient of Equation 1 with respect to a shape parameter $\beta_t$ can be computed as:

$$\frac{\partial(\mathrm{HPSL}_{\mathrm{v}_{\varkappa}})}{\partial \beta_t} = \sum_i \omega_i \mathbf{C_{j_i}}(\mathbf{b_t^{v_{\varkappa}}} - \mathbf{b_0^{v_{\varkappa}}}) \quad \text{for } t = 1, 2, \ldots, 7$$

where, $\mathrm{HPSL}_{\mathrm{v}_{\varkappa}} = \Upsilon_{\mathrm{v}_{\varkappa}}(\Theta, \beta, \alpha)$ as defined in Equation 5 in the main paper. According to Equation 1, bones scales $\alpha$ influence the joints positions and vertices positions. Hence, the resultant gradient with respect to a hand scale $\alpha_s$, can be calculated as:

$$\frac{\partial(\textbf{HPSL})}{\partial \alpha_s} = \frac{\partial \mathbf{F}}{\partial \alpha_s} + \frac{\partial \Upsilon}{\partial \alpha_s} \qquad \text{for } s = 1, 2, \ldots, 6$$

To compute the partial derivative of $\mathbf{F}$ with respect to $\alpha_s$, we need to derivate $\mathrm{F}_{j_i}(\Theta, \alpha)$ (please refer to Equation 4 in the paper) with respect to its associated scale parameter $\alpha_s$. Hence,

$$\frac{\partial F_{j_i}}{\partial \alpha_s} = \sum_{k \in S_k} \Big[ \Big( \prod_{n \in S_{j_i}} [\mathrm{Rot}_{\phi_n}(\theta_n)] \times [\mathrm{Trans}(\alpha B)] \Big) [0, 0, 0, 1]^T \Big] \qquad (2)$$

where,

$$\mathrm{Trans}(\alpha B) = \begin{cases} \mathrm{Trans}(\alpha_n B_n) & \text{if } n \neq k \\ \mathrm{Trans}(\alpha_n B_n)' & \text{if } n = k \end{cases}$$

and, $S_k$ is the set of parent joints of $j_i$ that share the same scale parameter $\alpha_s$. $S_{j_i}$ is the set of joints along kinematic chain from $j_i$ to the root joint and $\phi$ is the rotation axis. In a similar way, the gradient of $\Upsilon_{\mathrm{v}_{\varkappa}}$ (please refer to Equation 5 in main paper) with respect to $\alpha_s$ can be computed as:

$$\begin{aligned} \frac{\partial \Upsilon_{\mathrm{v}_{\varkappa}}}{\partial \alpha_s} &= \sum_i \omega_i \frac{\partial \mathbf{C_{j_i}}}{\partial \alpha_s} \mathrm{v}_{\varkappa} \\ &= \sum_i \omega_i [\mathrm{M}_{j_i} (\mathrm{M}_{j_i}^{*-1})' + (\mathrm{M}_{j_i})' \mathrm{M}_{j_i}^{*-1}] \mathrm{v}_{\varkappa} \end{aligned}$$

The derivative of $\mathrm{M}_{j_i}(\Theta, \alpha)$ can be calculated by the following equation:

$$\mathrm{M}'_{j_i} = \sum_{k \in S_k} \Big[ \Big( \prod_{n \in S_{j_i}} [\mathrm{Rot}_{\phi_n}(\theta_n)] \times [\mathrm{Trans}(\alpha B)] \Big) \Big] \qquad (3)$$

where,

$$\mathrm{Trans}(\alpha B) = \begin{cases} \mathrm{Trans}(\alpha_n B_n) & \text{if } n \neq k \\ \mathrm{Trans}(\alpha_n B_n)' & \text{if } n = k \end{cases}$$

In order to calculate the derivative of $\mathrm{M}_{j_i}^{*-1}(\alpha)$, we can perform the following computation:

$$(\mathrm{M}_{j_i}^{*-1})' = -\mathrm{M}_{j_i}^{*-1} \mathrm{M}_{j_i}^{*\prime} \mathrm{M}_{j_i}^{*-1} \qquad (4)$$

Likewise, for the pose parameters $\Theta$, we compute the following equation:

$$\frac{\partial(\textbf{HPSL})}{\partial \theta_p} = \frac{\partial \mathbf{F}}{\partial \theta_p} + \frac{\partial \Upsilon}{\partial \theta_p} \qquad \text{for } p = 1, 2, \ldots, 26$$

Accordingly, the derivative of $\mathrm{F}_{j_i}(\Theta, \alpha)$ with respect to a pose parameter $\theta_p$, is simply to replace the rotation matrix of $\theta_p$ by its derivation as given by the following equation:

Figure 1: The virtual desktop setup used to generate the images.

$$\frac{\partial F_{j_i}}{\partial \theta_p} = \Big( \prod_{n \in S_{j_i}} [\mathrm{Rot}_\phi(\theta)] \times [\mathrm{Trans}(\alpha_n B_n)] \Big) [0,0,0,1]^T \tag{5}$$

where,

$$\mathrm{Rot}_\phi(\theta) = \begin{cases} \mathrm{Rot}_{\phi_n}(\theta_n) & \text{if } n \neq p \\ \mathrm{Rot}_{\phi_n}(\theta_n)' & \text{if } n = p \end{cases}$$

The derivative of $\Upsilon_{\mathrm{v}_\varkappa}$ with respect to $\theta_p$ can be computed as:

$$\begin{aligned} \frac{\partial \Upsilon_{\mathrm{v}_\varkappa}}{\partial \theta_p} &= \sum_i \omega_i \frac{\partial \mathbf{C_{j_i}}}{\partial \theta_p} \mathrm{v}_\varkappa \\ &= \sum_i \omega_i [(\mathrm{M}_{j_i})' \mathrm{M}_{j_i}^{*-1}] \mathrm{v}_\varkappa \quad \text{for } p = 1, 2, \ldots, 26 \end{aligned}$$

$\mathrm{M}_{j_i}'$ can be calculated by the following equation as:

$$\mathrm{M}_{j_i}' = \Big( \prod_{k \in S_{j_i}} [\mathrm{Rot}_\phi(\theta)] \times [\mathrm{Trans}(\alpha_k B_k)] \Big) \tag{6}$$

where,

$$\mathrm{Rot}_\phi(\theta) = \begin{cases} \mathrm{Rot}_{\phi_k}(\theta_k) & \text{if } k \neq p \\ \mathrm{Rot}_{\phi_k}(\theta_k)' & \text{if } k = p \end{cases}$$

## 2. Synthetic Dataset

In this section, we briefly discuss about the SynHand5M dataset generation.

We have set up a realistic desktop environment (Figure 1) by sitting the full character on an ergonomic chair, in front of an office desk on which lies a 27" monitor. As in most all-in-one PC configurations, the camera is embedded at the top-center of the monitor frame. In the default position, the hand palm faces the camera orthogonally and the fingers point up. The hand is positioned at 45cm from the camera. The center of the palm is aligned with the center of the camera view. Enforcing an ergonomically plausible posture for the whole character's body facilitates the coherent positioning of the elbow, thus leading to realistic wrist bending and forearm orientation. Our virtual camera simulates a Creative Senz3D Interactive Gesture Camera [2]. It renders images of resolution 320x240 using diagonal field of view of 74 degrees.

One of the main objectives of our synthetic dataset is to provide training data with a wide range of variation, both for poses for hand shapes, such that a neural network model can be trained to accurately estimate average cases as well as extreme ones. In general, achieving a good range of variation for the hand poses is not a challenge; for a synthetic database, the image generator must randomize the fingers rotation of a min/max range of rotation, while for a real-world database subjects must over-articulate their motion in front of a camera. However, the variation in shape is more challenging for real-world databases, where a significant variability may not be present in a given cohort of human users. For example, the BigHand2.2M [8] database was captured from 10 users, and the MANO [7] database was built from the contribution of 31 users.

For this reason, for the generation SynHand5M hand shapes we rely of an artistic-driven body (and hand) generator. SynHand5M uses the hand model generated by ManuelBastionLAB [4], which is a procedural full-body generator distributed as add-on of the Blender [1] 3D authoring software. Without constraints the hand generator can easily lead to impossible hand shapes, so we tune the ranges of the generation parameters using real-world statistical data from the DINED [5] anthropometric database. DINED is a repository collecting the results of several anthropometric databases, including the CAESAR surface anthropometry survey [6].

In the hand shape generator, the shape of the hand can be modulated on 7 hand shape parameters, namely: *Length*, *Mass*, *Size*, *Palm Length*, *Fingers Inter-distance*, *Fingers Length* and *Fingers Tip-Size*. In order to define realistic range limits, we extracted the average and standard deviation (sd) of the size of the hand of caucasian males from two DINED indices: (43) Hand Length, mean=183mm, sd=14; and (44) Palm Width, mean=83mm, sd=8. Then, we manually tuned the ranges of the *Hand-Size* parameters in order to cover the measured means $\pm$ 3sd (99% of the population). We first manually determined the min/max values of *Size*, *Mass*, and *InterDist* to match Palm Width (44) $\pm$ 3sd. Then, we determined the min/max values for *Length*, *PalmLength*, and *FingersLength* in order to match Hand Length (43) $\pm$ 3sd. Finally, since precise statistical data are unavailable for the *FingersTipSize*, we

subjectively limited its range to $[0.2, 0.8]$ in order to avoid too unrealistic aspects.

Since the 7 hand generation parameters accumulate in offsetting the same mesh vertices, in spite of the given constraints some parameters combination can lead to hand shapes beyond any statistical limits. However, in this context, the relative drop in realism is still acceptable because extreme hand shapes (although they will never match a real-world input) help the randomization to produce as much variability as possible and cover border-line cases.

The SynHand5M is generated by randomly sampling from the parameters, which are divided in three categories: hand pose, shape, and view point, and rendering the view from a virtual camera. To modulate the hand pose, we manipulate the 26 DoFs of our hand model; see Figure 3(b) in the main paper. For each finger, rotations are applied to flexion of all phalanges plus the abduction of the proximal phalanx. Additionally, in order to increase the realism of the closed fist configuration, the roll of middle, ring, and pinky fingers is derived from the abduction angle of the same phalanx. The rotation limits of the fingers have been set to bring the hand from a closed fist to an over-extended aperture, respecting anatomical constraints and avoiding the fingers to enter the palm. However, for some combinations, the rotation angles lead to inter-penetration between the thumb and one of the other fingers. Inter-penetrating configurations are automatically discarded from the dataset if a collision of the fingers' geometry occurs. In the default position, the hand palm faces the camera orthogonally and the fingers point up. The hand can rotate about three DoFs: roll around its longitudinal axis (i.e. along the forearm), rotate around the palm orthogonal axis (i.e. rolling in front of the camera), and rotate around its transversal axis (i.e. flexion/extension of the wrist).

The SynHand5M database is divided in chunks of 100,000 images. Each chunk is stored in a different directory. Each chunk comes with a dataframe in CSV (Comma-separated values) text format. Each line of the dataframe corresponds to a hand *configuration* (i.e., modulation of finger rotation, hand size, and hand rotation)

The dataframe columns report:

- The hand root $x, y, z$ position in space.
- The hand root rotation in space, in both Euler-angles and quaternion format.
- The hand shape parameters.
- For each hand bone: the rotation relative to the parent bone (Euler angles), the location in space of the two bone extremities.

All of the above information are sufficient to re-generate the dataset images at different resolution. An additional information file contains details about the camera transformation and projection matrices, allowing for a conversion between 3D and pixel spaces.

For each configuration, the dataset contains three additional files:

- The depth image is saved as 16-bit gray-scale PNG file, where the gray value (in $[0, 65535]$) is the distance in mm from the camera sensor.
- The vertex coordinates are saved as $x, y, z$ float triplets in a binary file.
- The colored segmentation of the hand is produced by performing an RGB rendering of the hand on which a manually painted texture is applied. The texture distinguishes between palm and phalanges. Segmentation images are saved as 24-bit true color PNG images.

## References


[1] Blender. https://www.blender.org, March 2018. 2

[2] Creative. Senz3d interactive gesture camera. https://us.creative.com/p/web-cameras/creative-senz3d, March 2018. 2

[3] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 165–172. ACM Press/Addison-Wesley Publishing Co., 2000. 1

[4] ManuelBastioni. v1.5.0. http://www.manuelbastioni.com, March 2018. 2

[5] J. Molenbroek. Dined, anthropometric database. https://dined.io.tudelft.nl/, 2004. 2

[6] K. Robinette, H. Daanen, and E. Paquet. The CAESAR project: a 3-D surface anthropometry survey. pages 380–386. IEEE Comput. Soc, 1999. 2

[7] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):245:1–245:17, Nov. 2017. 2

[8] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2605–2613. IEEE, 2017. 2