HandVoxNet: Deep Voxel-Based Network for 3D Hand Shape and Pose Estimation from a Single Depth Map — Supplementary Material —

Jameel Malik[1,2,3] Ibrahim Abdelaziz[1,2] Ahmed Elhayek[2,4] Soshi Shimada[5]
Sk Aziz Ali[1,2] Vladislav Golyanik[5] Christian Theobalt[5] Didier Stricker[1,2]

[1]TU Kaiserslautern [2]DFKI Kaiserslautern [3]NUST Pakistan [4]UPM Saudi Arabia [5]MPII Saarland

In this document, we provide the network details of HandVoxNet (Sec. 1 and 2). Also, we show qualitative results of the synthesizers, which reconstruct voxelized depth maps from the shape representations (Sec. 1.3).

1. Network Design

In this section, we describe in detail the architectures of the V2V-ShapeNet, V2S-Net, V2V-SynNet, S2V-SynNet, and DispVoxNet.

1.1. V2V-ShapeNet Architecture

V2V-ShapeNet regresses $\hat{\mathcal{V}}_S$ which is $64 \times 64 \times 64$ voxelized representation of hand shape, from input $\mathcal{I}_S$ (*i.e.,* $(\mathrm{N}+1)\ 44 \times 44 \times 44$ voxelized grids). Since V2V-ShapeNet learns to estimate a dense 3D hand shape representation from sparse 3D hand joints and depth map, it can be therefore considered as a decoder which tries to reconstruct voxelized hand shape as close as possible to ground truth $\mathcal{V}_S$. V2V-ShapeNet establishes a one-to-one mapping between the voxelized hand shape, voxelized depth map, and 3D joints heatmaps. Table 1 shows the architectural details of the 3D convolutions based V2V-ShapeNet. For weak supervision, V2V-SynNet reconstructs the voxelized depth map from the estimated voxelized hand shape representation (see Fig. 2 in the main paper). The samples of the reconstructed voxelized depth maps of NYU [4] and BigHand2.2M [6] real benchmarks are shown in Fig. 1(a).

1.2. V2S-Net Architecture

V2S-Net regresses K 3D hand mesh vertices $\hat{\mathcal{V}}_T$ from the input $\mathcal{I}_S$. The architecture of V2S-Net also consists of 3D convolutions, except for the last two layers that are fully connected (FC). Table 2 shows the architectural details of V2S-Net. Since V2S-Net regresses 3D coordinates of the shape, it does not establish a one-to-one mapping between the voxelized depth map and 3D joint heatmaps. For weak supervision, S2V-SynNet reconstructs voxelized depth maps from the estimated hand meshes. The samples of reconstructed voxelized depth maps are shown in Fig. 1(b).

ID	Layer	Output Sz	Kernel Sz	Stride/Padding	+
1	Input	(N+1) 44x44x44	-	-/-	-
2	3D Conv, BN, ReLU	(22) 44x44x44	7x7x7	1/3	-
3	3D Conv, BN, ReLU	(24) 38x38x38	7x7x7	1/0	-
4	3D Conv, BN, ReLU	(26) 32x32x32	7x7x7	1/0	-
5	3D Conv, BN, ReLU	(26) 32x32x32	3x3x3	1/1	-
6	3D Conv, BN	(26) 32x32x32	3x3x3	1/1	4
7	ReLU	(26) 32x32x32	-	-/-	-
8	3D DeConv, BN, ReLU	(8) 64x64x64	2x2x2	2/0	-
9	3D Conv, BN, ReLU	(8) 64x64x64	3x3x3	1/1	-
10	3D Conv, BN	(8) 64x64x64	3x3x3	1/1	8
11	ReLU	(8) 64x64x64	-	-/-	-
12	3D Conv, Sigmoid	(1) 64x64x64	1x1x1	1/0	-

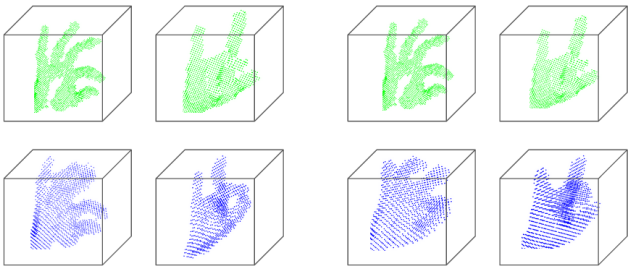
Table 1: V2V-ShapeNet architecture details. Output **Sz** consists of the number of channels and their spatial size. In the "+" column, the output of layer ID is added to the current layer's output in a voxel-wise manner.

ID	Layer	Output Sz	Kernel Sz	Stride/Padding	+
1	Input	(N+1) 44x44x44	-	-/-	-
2	3D Conv, BN, ReLU	(22) 44x44x44	7x7x7	1/3	-
3	3D MaxPooling	(22) 22x22x22	2x2x2	2/0	-
4	3D Conv, BN, ReLU	(22) 22x22x22	3x3x3	1/1	-
5	3D Conv, BN	(22) 22x22x22	3x3x3	1/1	3
6	ReLU	(22) 22x22x22	-	-/-	-
7	3D Conv, BN	(16) 22x22x22	1x1x1	1/0	-
8	3D Conv, BN, ReLU	(16) 22x22x22	3x3x3	1/1	-
9	3D Conv, BN	(16) 22x22x22	3x3x3	1/1	7
10	ReLU	(16) 22x22x22	-	-/-	-
11	3D Conv, BN, ReLU	(8) 22x22x22	1x1x1	1/0	-
12	3D MaxPooling	(8) 11x11x11	2x2x2	2/0	-
13	3D Conv, BN, ReLU	(1) 11x11x11	1x1x1	1/0	-
14	Flatten	11*11*11	-	-/-	-
15	FC, ReLU	400	-	-/-	-
16	FC	K*3	-	-/-	-

Table 2: V2S-Net architecture details.

1.3. V2V-SynNet and S2V-SynNet Architectures

V2V-SynNet and S2V-SynNet act as sources of weak supervision during the training phase and are not included in



(a) (b)

Figure 1: Samples of synthesized voxelized depth maps of NYU [4] and BigHand2.2M [6] datasets from V2V-SynNet (a) and S2V-SynNet (b). The first and second rows show the ground truth and the reconstructions, respectively.

the testing phase. These synthesizers reconstruct voxelized depth maps $\hat{V}_D$ from the hand shape representations. The details of the architectures of V2V-SynNet and S2V-SynNet are provided in Tables 3 and 4, respectively.

ID	Layer	Output Sz	Kernel Sz	Stride/Padding	+
1	Input	(1) 64x64x64	-	-/-	-
2	3D Conv, BN, ReLU	(8) 64x64x64	7x7x7	1/3	-
3	3D MaxPooling	(8) 32x32x32	2x2x2	2/0	-
4	3D Conv, BN	(16) 32x32x32	1x1x1	1/0	-
5	3D Conv, BN, ReLU	(16) 32x32x32	3x3x3	1/1	-
6	3D Conv, BN	(16) 32x32x32	3x3x3	1/1	4
7	ReLU	(16) 32x32x32	-	-/-	-
8	3D DeConv, BN, ReLU	(12) 38x38x38	7x7x7	1/0	-
9	3D Conv, BN, ReLU	(12) 38x38x38	3x3x3	1/1	-
10	3D Conv, BN	(12) 38x38x38	3x3x3	1/1	8
11	ReLU	(12) 38x38x38	-	-/-	-
12	3D DeConv, BN, ReLU	(8) 44x44x44	7x7x7	1/0	-
13	3D Conv, BN, ReLU	(8) 44x44x44	3x3x3	1/1	-
14	3D Conv, BN	(8) 44x44x44	3x3x3	1/1	12
15	ReLU	(8) 44x44x44	-	-/-	-
16	3D Conv, BN, ReLU	(8) 44x44x44	1x1x1	1/0	-
17	3D Conv, BN, ReLU	(8) 44x44x44	1x1x1	1/0	-
18	3D Conv, Sigmoid	(1) 44x44x44	1x1x1	1/0	-

Table 3: V2V-SynNet architecture details.

ID	Layer	Output Sz	Kernel Sz	Stride/Padding	+
1	Input	K*3	-	-/-	-
2	FC, ReLU	400	-	-/-	-
3	Reshape	(400) 1x1x1	-	-/-	-
4	3D DeConv, BN, ReLU	(128) 3x3x3	3x3x3	1/0	-
5	3D DeConv, BN, ReLU	(64) 6x6x6	3x3x3	2/1	-
6	3D DeConv, BN, ReLU	(32) 11x11x11	6x6x6	1/0	-
7	3D DeConv, BN, ReLU	(16) 22x22x22	3x3x3	2/1	-
8	3D Conv, BN, ReLU	(8) 44x44x44	1x1x1	1/0	-
9	3D Conv, BN, ReLU	(8) 44x44x44	1x1x1	1/0	-
10	3D Conv, Sigmoid	(1) 44x44x44	1x1x1	1/0	-

Table 4: S2V-SynNet architecture details.

1.4. DispVoxNet Architecture

In contrast to the original DispVoxNets [3] composed of the displacement estimation and refinement stages, we replace the refinement stage with Laplacian smoothing [5].

ID	Layer	Output Sz	Kernel Sz	Stride/Padding	⊕
1	Input	(2) 64x64x64	-	-/-	-
2	3D Conv	(8) 64x64x64	7x7x7	1/3	-
3	LeakyReLU	(8) 64x64x64	-	-/-	-
4	3D MaxPooling	(8) 32x32x32	2x2x2	2/0	-
5	3D Conv	(16) 32x32x32	5x5x5	1/2	-
6	LeakyReLU	(16) 32x32x32	-	-	-
7	3D MaxPooling	(16) 16x16x16	2x2x2	2/0	-
8	3D Conv	(32) 16x16x16	3x3x3	1/1	-
9	LeakyReLU	(32) 16x16x16	-	-/-	-
10	3D MaxPooling	(32) 8x8x8	2x2x2	2/0	-
11	3D Conv	(64) 8x8x8	3x3x3	1/1	-
12	LeakyReLU	(64) 8x8x8	-	-/-	-
13	3D Deconv	(64) 16x16x16	2x2x2	2/0	10
14	3D Deconv	(64) 16x16x16	3x3x3	1/1	-
15	LeakyReLU	(64) 16x16x16	-	-/-	-
16	3D Deconv	(32) 32x32x32	2x2x2	2/0	7
17	3D Deconv	(32)32x32x32	5x5x5	1/2	-
18	LeakyReLU	(32) 32x32x32	-	-/-	-
19	3D Deconv	(16) 64x64x64	2x2x2	2/0	4
20	3D Deconv	(16) 64x64x64	7x7x7	1/3	-
21	LeakyReLU	(16) 64x64x64	-	-/-	-
22	3D Deconv	(3) 64x64x64	3x3x3	1/1	-

Table 5: DispVoxNet architecture details. The " " column marks layer ID whose outputs are concatenated and passed as input to the current layer. The negative slope for LeakyReLU is 0.01.

By doing so, we simplify the pipeline and avoid the training of another instance of DispVoxNet in the refinement stage. We follow the original network architecture of [3], see Table 5. DispVoxNet accepts two voxel grids ($\hat{\mathcal{V}}_S$ and $\hat{\mathcal{V}}'_T$)) and returns voxel displacements to register voxelized shaps. After applying the estimated displacements on the template, we apply Laplacian smoothing on it to reduce the shape roughness, and obtain the final hand shape. We train DispVoxNet in a supervised manner. However, the ground truth displacements are not available and obtaining them between the voxel and surface shapes is not straightforward. To circumvent this problem, we use the displacements between the shape surface and the corresponding ground truth shape in SynHand5M dataset. This is possible because the shape surface generated by V2S-Net preserves the topology and the number of mesh vertices during the training.

2. NRGA-Based Registration

We provide more details on the modification of the nearest neighbors rule mentioned in Sec. 4.1 (main matter). To highlight the role of this modification, we summarize the shape deformation and the optimization scheme of NRGA.

Optimization Method. Given the estimated hand shape surface $\hat{\mathcal{V}}_T$ and the voxelized shape $\hat{\mathcal{V}}_S$, NRGA defines the total gravitational potential energy (GPE) of the system as

$$\mathbf{E}(\mathbf{R};\mathbf{t}) = \sum_{k=1}^{K} \sum_{j\,2\ \hat{v}_k^S} \frac{!_k}{\left(k\mathbf{R}_k\hat{V}_T^k + \mathbf{t}_k \quad \hat{V}_S^j k + \right)}; \qquad (1)$$

which is the weighted sum of the inverse of the Euclidean distances between the mesh vertices $\hat{\mathcal{V}}_T$ =

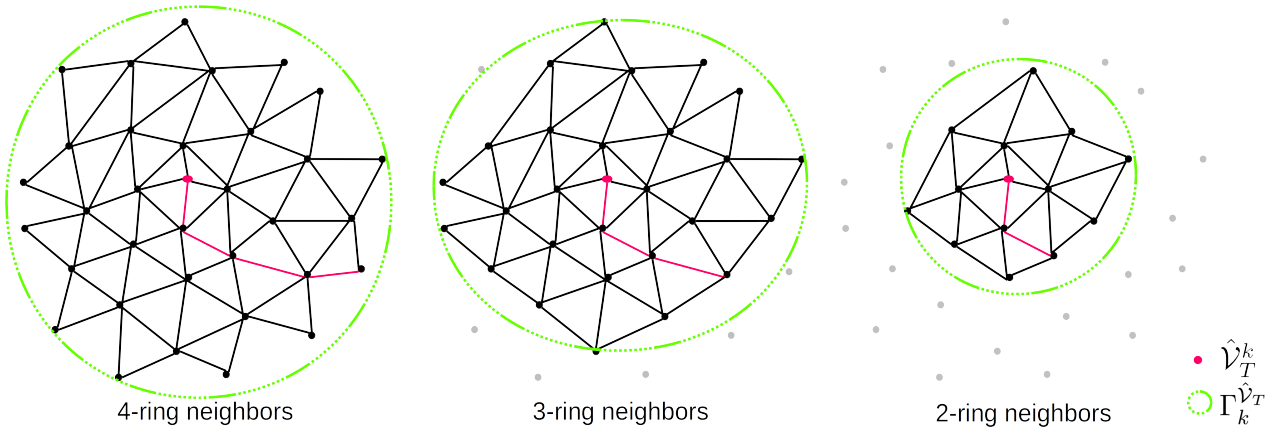


Figure 2: Selection of interacting vertices in NRGA. A vertex $\hat{V}_T^k$ in red selects either 4-ring, 3-ring or 2-ring neighbourhood vertices to define a local subspace of our deformable template $\hat{V}_T$. The vertices in black are enclosed inside the region $\Gamma_k^{\hat{V}_T}$ and reachable from $\hat{V}_T^k$ with the shortest path-length n for n-ring neighbours.

$[\hat{\mathcal{V}}_T^1, \hat{\mathcal{V}}_T^2, \ldots, \hat{\mathcal{V}}_T^K]$ and their neighbouring lattice vertices from the voxelized hand $\hat{\mathcal{V}}_S = [\hat{\mathcal{V}}_S^1, \hat{\mathcal{V}}_S^2, \ldots, \hat{\mathcal{V}}_S^M]$, with $\|.\|$ denoting ℓ_2-norm and force softening length ϵ. In the definition of GPE (1), ω_k is a product of the *gravitational constant* G and the masses of the interacting vertex pair $(\hat{\mathcal{V}}_T^k, \hat{\mathcal{V}}_S^j)$. The number of vertices in the template hand shape is fixed to $K = 1193$. On the other hand, only the lattice points with output probabilities ≥ 0.8 are selected to represent $\hat{\mathcal{V}}_S$. This results in a varying number of total vertices M in $\hat{\mathcal{V}}_S$ for different input samples. The GPE (1) is minimized to estimate the optimum transformation parameters, *i.e.* K rotations $\mathbf{R} = [\mathbf{R}_1, \ldots, \mathbf{R}_K]$ and translations $\mathbf{t} = [\mathbf{t}_1, \ldots, \mathbf{t}_K]$ for each hand shape vertex. By applying the estimated rigid transformations, $\hat{\mathcal{V}}_T$ is deformed to match with the underlying shape of $\hat{\mathcal{V}}_S$. NRGA requires k-d trees to be built independently on source $\hat{\mathcal{V}}_T$ and target $\hat{\mathcal{V}}_S$ which help to obtain the nearest neighbours of every source vertex $\hat{\mathcal{V}}_T^K$. The number of nearest neighbours are fetched from $\hat{\mathcal{V}}_T^k$ and $\hat{\mathcal{V}}_S$ as a proportion $\rho \in [0.02 - 0.1]\%$ of the total number of points. These neighbours form sets of local regions $\{\Gamma_k^{\hat{\mathcal{V}}_T}\}$ and $\{\Gamma_k^{\hat{\mathcal{V}}_S}\}$ for the template and reference point clouds, respectively. The vertices in $\Gamma_k^{\hat{\mathcal{V}}_S}$ appear as lattice corners, whereas vertices in $\Gamma_k^{\hat{\mathcal{V}}_T}$ are ***not*** selected as a portion of nearest neighbours, and instead as a set of vertices inside the 4-path distance from $\hat{\mathcal{V}}_T^k$ as shown in Fig. 2.

Optimization Parameters. We set the parameters of NRGA used in our HandVoxNet as follows: $G = 0.667$, masses $m(\hat{\mathcal{V}}_T) = m(\hat{\mathcal{V}}_S) = 1.0$ of all point vertices in $\hat{\mathcal{V}}_T$ and $\hat{\mathcal{V}}_S$, $\epsilon = 0.2$, $\rho = 0.02$, energy dissipation rate $\eta = 0.2$ and time integration step $\Delta t = 0.006$.

3. More Qualitative Results

We present more qualitative results of 3D hand mesh reconstruction for NYU [4] and BigHand2.2M [6] test datasets (as shown in Fig. 3 and 4, respectively). We demonstrate that our method estimates visually more accurate hand shapes for NYU dataset compared to the previous works [1, 2]. Our results on selected test samples of the NYU dataset have been made publicly available for comparisons[1].

References

[1] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Heloir, and Didier Stricker. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *International Conference on 3D Vision (3DV)*, 2018. 3, 4

[2] Jameel Malik, Ahmed Elhayek, and Didier Stricker. Whspnet: A weakly-supervised approach for 3d hand shape and pose recovery from a single depth image. *Sensors*, 19(17):3784, 2019. 3, 4

[3] Soshi Shimada, Vladislav Golyanik, Edgar Tretschk, Didier Stricker, and Christian Theobalt. Dispvoxnets: Non-rigid point set alignment with supervised learning proxies. In *International Conference on 3D Vision (3DV)*, 2019. 2

[4] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5):169, 2014. 1, 2, 3, 4

[5] Jörg Vollmer, Robert Mencl, and Heinrich Mueller. Improved laplacian smoothing of noisy surface meshes. In *Computer Graphics Forum*, pages 131–138, 1999. 2

[1] https://cloud.dfki.de/owncloud/index.php/s/YfWw3SN92s79N4L