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

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

Hybrid system

Problem

Directly regress 3D hand meshes from 2D depth images via 2D convolutional neural networks cause perspective distortions

Solution

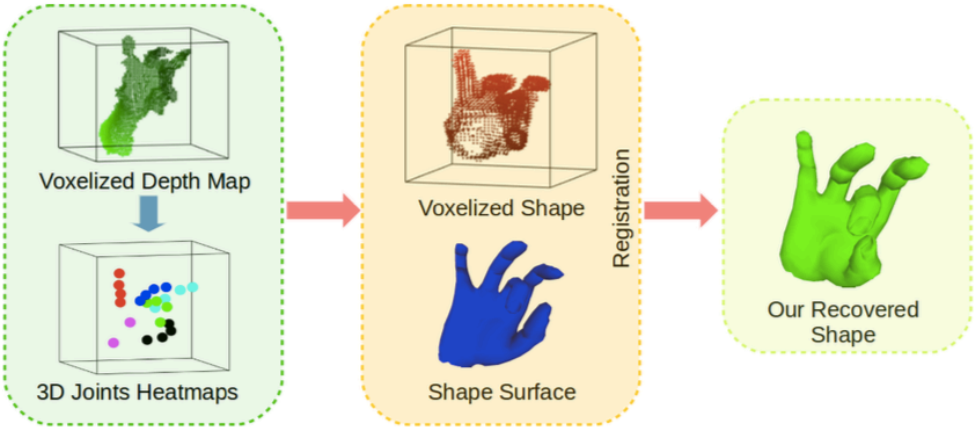
Input 3D voxelized depth map, input depth image is converted into a voxelized grid (*i.e.,* VD) of size 88 × 88 × 88, by using intrinsic camera parameters and a fixed cube size

Using 2 hand shape representations:

3D voxelized grid of the shape

3D hand surface

Combine from registering the hand surface to the voxelized hand shape



Output

N 3D hand joint locations J ∈ R3×N (i.e., 3D pose)

K = 1193 3D vertex locations V ∈ R3×K (i.e., 3D shape).

Solution

Register the estimated shape by V2S-Net to the probabilistic shape representation estimated by FCN (V2V-ShapeNet) using DispVoxNets pipeline

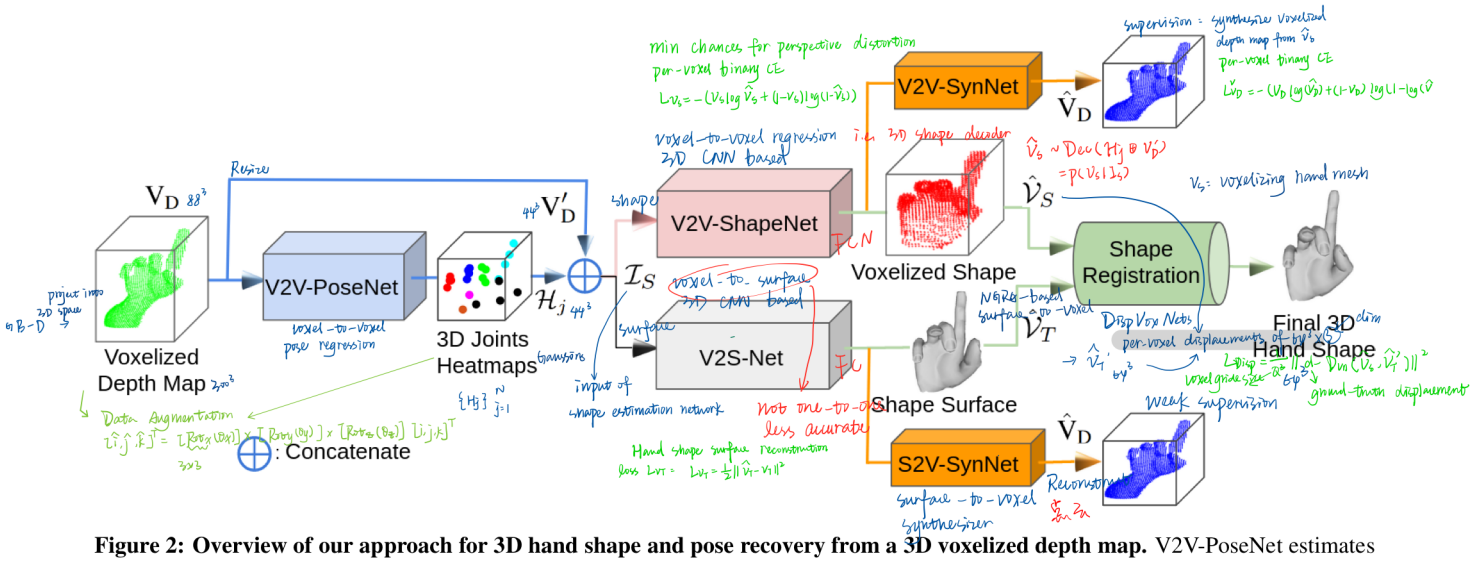


Figure 2: Overview of our approach for 3D hand shape and pose recovery from a 3D voxelized depth map. V2V-PoseNet estimates 3D joints heatmaps (*i.e.*, pose). Hand shape is obtained in two phases. First, V2V-ShapeNet and V2S-Net estimate the voxelized shape and shape surface, respectively. Thereby, V2V-SynNet and S2V-SynNet synthesize the voxelized depth acting as sources of weak-supervision. They are excluded during testing. In the second phase, shape registration accurately fits the shape surface to the voxelized shape.

	Good point	Drawback
V2V-ShapeNet(FC)	Estimate hand shapes while preserving the order and number of points	Local special info loss
V2S-Net(FCN)	Geometry regression	Inconsistent number of points and loses point order.

Data Augmentation in 3D

Originally treat depth maps as 2D data; The representation of the depth map in voxelized form makes it convenient to perform data augmentation in all three dimensions.

Understanding Human Hands in Contact at Internet Scale

Dandan Shan[1], Jiaqi Geng*[1], Michelle Shu*[2], David F. Fouhey[1]
[1]University of Michigan, [2]Johns Hopkins University
{dandans,jiaqig,fouhey}@umich.edu, msh1@jhu.edu

Detect hand box in RGB that enables mesh reconstruction systems => hand detections
数据采集
Gathering implicit video dataset

identifying an overcomplete set of candidate videos using generic queries and filtering out irrelevant videos.

Annotation: For every hand in each image, we obtained the following annotations: (a) a bounding box around the hand; (b) side: left / right, which is crucial for mesh re- construction; (c) the hand contact state ({no contact, self-

contact, other person contact, in contact with portable object, in contact with a non-portable object}), which provides insights into what the person is doing; and (d) a bounding box around the object the person is contacting *irrespective of name*.

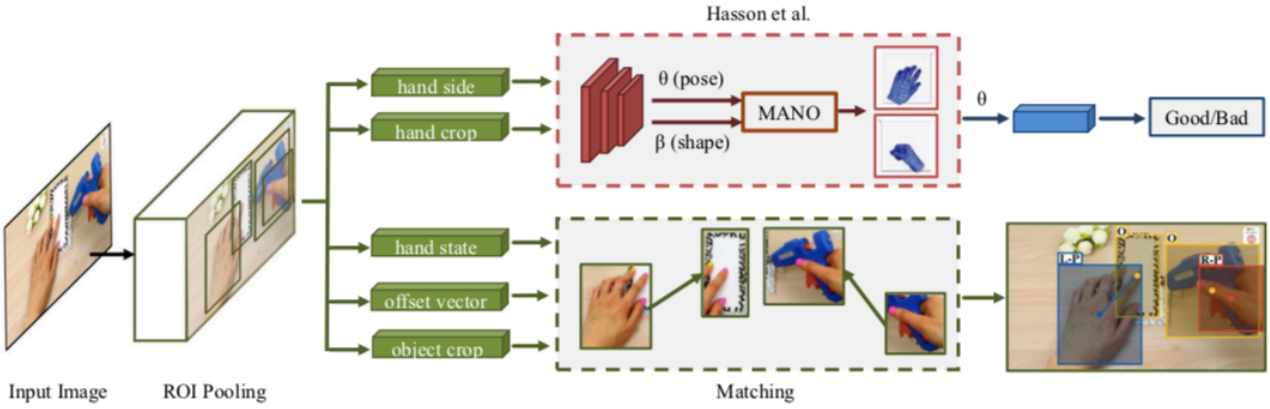


Figure 3. Our system can act as a foundation to understand interacting hands on the Internet. Our system takes a single RGB image and detects hands (irrespective of scale) and for every hand predicts: a box, side, contact state, and a box around the object it is touching. We can then (1) obtain a parse of hand state; and (2) use the hand box and side to feed a reconstruction system like [21]. To help make better use of Internet reconstructions, we introduce a self-supervised system that assesses mesh quality that we train on our data.

Weakly-Supervised Mesh-Convolutional Hand Reconstruction in the Wild

Dominik Kulon[1, 3] Riza Alp Güler [1, 3]
Iasonas Kokkinos [3] Michael Bronstein [1, 2, 4] Stefanos Zafeiriou [1, 3]
[1]Imperial College London [2]USI Lugano [3]Ariel AI [4]Twitter
{d.kulon17, r.guler, m.bronstein, s.zafeiriou}@imperial.ac.uk iasonas@arielai.com

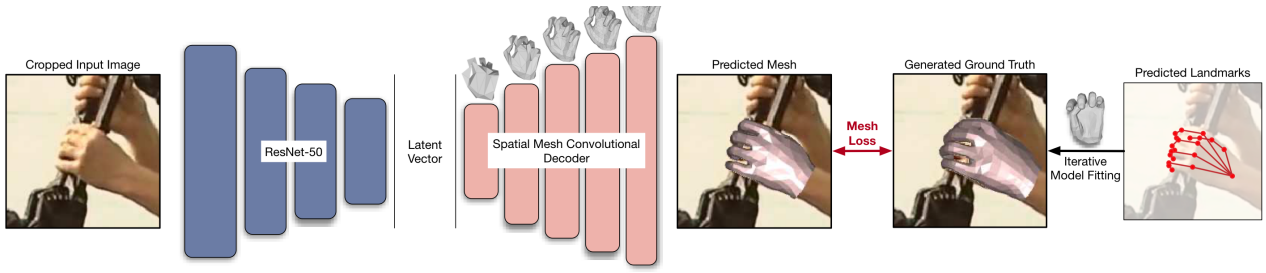


Figure 1: We propose an approach for end-to-end neural network training with mesh supervision that is obtained through an automated data collection method. We process a large collection of YouTube videos and analyze them with 2D hand keypoint detector followed by parametric model fitting (right side). The fitting results are used as a supervisory signal ('mesh loss') for a feed-forward network with a mesh convolutional decoder tasked with recovering a 3D hand mesh at its output (left side).

AUTOMATED DATASET GENERATION FOR 3D RECONSTRUCTION AND POSE ESTIMATION

Detect key points using OpenPose

Lift key points into **3D shapes** by iteratively fitting a **3D deformable model**

3D Shape Representation

$$M(\beta, \theta, \vec{T}_\delta, s; \phi) : \mathbb{R}^{|\beta| \times |\theta| \times |\vec{T}_\delta| \times |s|} \rightarrow \mathbb{R}^{N \times 3} \quad (1)$$

$$\theta_i = P(w)_i = \frac{\sum_{c=1}^{C} \exp(w^c) P_i^c}{\sum_{c=1}^{C} \exp(w^c)}.$$

pre-computed cluster(2)centres

Total #clusters $C = 64$

Represent all constrained angles in terms of w in $R^{K \times C}$

Does not model pairwise dependencies => future work

Requires only a small dataset of angles

Parametric Model Fitting

Regress a matching 3D pose ($K = 16$ joint and $F = 5$ fingertip positions) from MANO mesh #N vertices

$$J(\beta, w, \vec{T}_\delta, s) = \mathcal{J}^T M(\beta, P(w), \vec{T}_\delta, s). \quad (3$$

$N \times 3$, $N \times (K+F)$

Fit the model to 2D annotation by objective:

$$\{\beta^*, w^*, \vec{T}_\delta^*, s^*\} = \arg \min_{\beta, w, \vec{T}_\delta, s} (E_{2D} + E_{bone} + E_{reg}), \quad (4)$$

E_{2D}: 2D projection; E_{bone}: length preservation; E_{reg}: regularization

used for weak supervision – ground truth

2D landmarks extracted by OpenPose

HAND RECONSTRUCTION NETWORK

Spiral Operator

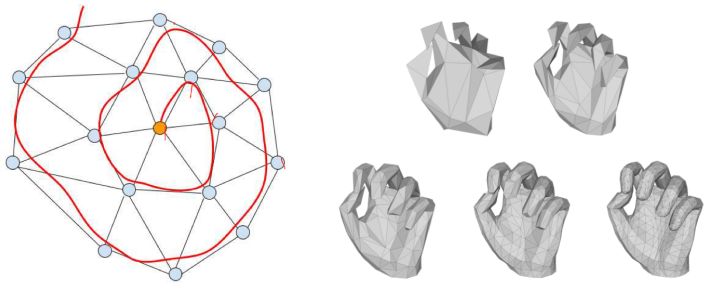
Only consider k=2 here.

$$\begin{aligned} 0\text{-ring}(v) &= \{v\}, \\ (k+1)\text{-ring}(v) &= N(k\text{-ring}(v)) \setminus k\text{-disk}(v), \\ k\text{-disk}(v) &= \cup_{i=0\ldots k}\, i\text{-ring}(v), \end{aligned} \tag{7}$$

$$\mathrm{S}(v) = (v, 1\text{-ring}(v), \ldots, k\text{-ring}(v)). \tag{8}$$

$$(f * g)_v = \sum_{\ell=1}^{L} g_\ell\, f\big(S_\ell(v)\big), \tag{9}$$

1st neighbour, kernel, spiral cow, spiral length, fixed in each layer



a) b)

Figure 2: a) Spiral selection of neighbours for a center vertex in clockwise order (taken from [6]). b) The hierarchy of topologies used in our decoder.

Down sampling

#vertices reduced by the factor of 2

Contract vertex pairs based on **quadric error metrics**,

1 Project each collapsed node $v_q \in V$ into the closest triangle $v_i, v_j, v_k \in V_d$ in the downsampled mesh V_d obtaining $\tilde{v}_p$.

2 Use the barycentric coordinates w_1, w_2, w_3 such that $\tilde{v}_p = w_1 v_i + w_2 v_j + w_3 v_k$ and $w_1 + w_2 + w_3 = 1$ of the projected vertex to define interpolation weights for the upsampling matrix $Q_u \in R_{m \times n}$,

Up sampling

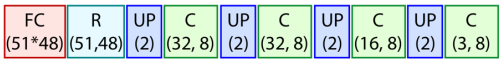
$V_u = Q_u V_d$ and $m > n$ hold, is formed by setting $Q_u(q,i) = w_i$, $Q_u(q,j) = w_j$, $Q_u(q,k) = w_k$, and $Q_u(q,l) = 0\ for\ l\ not \in \{i,j,k\}$.

The number of vertices at each layer $\mathrm{n} \in \{51, 100, 197, 392, 778\}$.

Architecture Encoder-Decoder

Image crop X

=> latent vector $Z = E_{image}(X)$; encoder E_{image} ResNest-50

=> mesh $y = D_{mesh}(Z)$; spiral decoder D_{mesh} 

, leaking ReLU as activation function; Decode a mesh directly from the image encoding and apply spatial convolutions with pooling layers

1 Learn invariant shape features in a triangular mesh

2 Apply spiral filters to generate hands directly from an image encoding (spatial convolution)

Training

$$\mathcal{L} = \lambda_{vertex}|\hat{\mathcal{Y}} - \mathcal{Y}|_1 + \lambda_{edge} \sum_{(u,v)\in\mathcal{E}_{mesh}} | \, ||\hat{\mathcal{Y}}_v - \hat{\mathcal{Y}}_u|| - ||\mathcal{Y}_v - \mathcal{Y}_u|| \, | \quad (10)$$

No explicit pose estimation loss; joint coordinates are obtained from eq.3

The network is trained with Adam optimizer

Adapts HOI domain to the single-HPE domain in **image space** rather than feature space via 2 **image generation** methods:

1 2D monocular guidance by **GAN** (Generative adversarial network) => align hands

2 3D mesh guidance by **MR** (mesh renderer) using estimated 3D meshes and textures => fill in occluded pixels

Thus 1) 3D HPE more accurate 2) HOI input images are translated to segmented and de-occluded hand-only images => more accurate

The pipeline is trained by hand-only images with pose labels and HOI images without pose labels Simultaneously:122

1 domain adaptation: weak supervision by 2D object segmentation masks and 3D pose (hand-only) labels

2 HPE

Problem: lack HOI 3D annotations

Solution: training a new HPE under the HOI scenario by mapping input HOI image to the corresponding hand-only image thus use only datasets:

1 Input RGB in hand-only and HOI

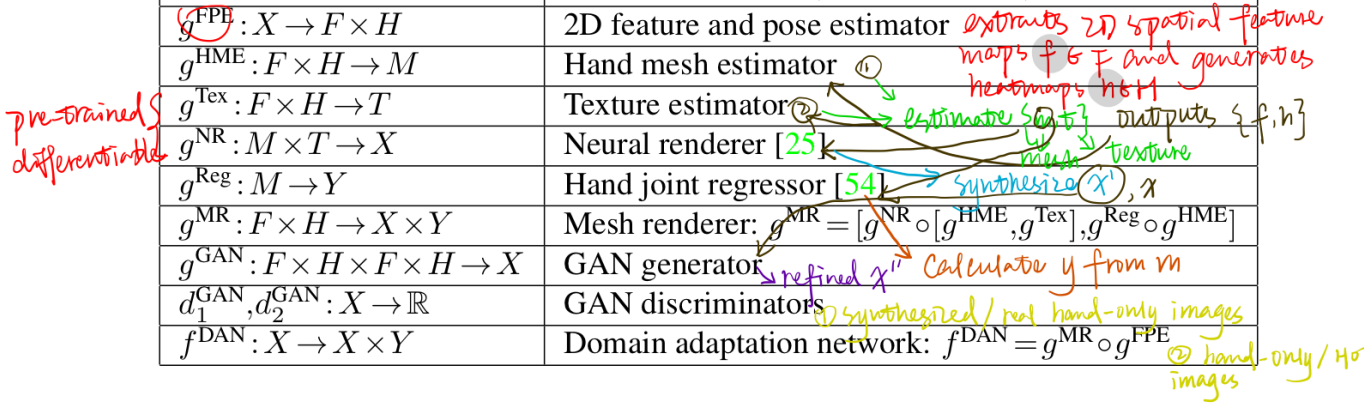
2 Skeleton annotations for hand-only

3 2D binary segmentation masks for hand-only and HOI (extracted based on depth maps)

⇨ Require restoring occluded regions

$D^{\mathrm{R}}_{\mathrm{Hand}}=\{(\mathbf{x},\mathbf{s}_{\mathrm{Hand}},\mathbf{y})\}$	Real hand-only data (*STB*)
$D^{\mathrm{S}}_{\mathrm{Hand}}=\{(\mathbf{x},\mathbf{s}_{\mathrm{Hand}},\mathbf{y})\}$	Synthetic hand-only data (SynthHands, *RHD*)
$D^{\mathrm{R}}_{\mathrm{HOI}}=\{(\mathbf{x},\mathbf{s}_{\mathrm{HOI}})\}$	Real HOI data (*CORe50*)
$D^{\mathrm{S}}_{\mathrm{Paired}}=\{(\mathbf{x},\mathbf{x}_*,\mathbf{s}_{\mathrm{Hand}},\mathbf{s}_{\mathrm{HOI}}\}$	Paired synthetic HOI ($\mathbf{x}$) and hand-only ($\mathbf{x}_*$) images (*Obman*)
$D_{\mathrm{Hand}}=[D^{\mathrm{R}}_{\mathrm{Hand}},D^{\mathrm{S}}_{\mathrm{Hand}}]$	Hand-only data
$D=[D_{\mathrm{Hand}},D^{\mathrm{R}}_{\mathrm{HOI}},D^{\mathrm{S}}_{\mathrm{Paired}}]$	All training data that we use

$X\subset\mathbb{R}^{256\times256\times3}$	RGB images ($\mathbf{x}$: input; $\mathbf{x}'$: rendered by g^{MR}; $\mathbf{x}''$: synthesized by g^{GAN}; $\mathbf{z}$: final mesh estimate rendered by g^{MR})
$Y\subset\mathbb{R}^{21\times3}$	3D skeletal pose space
$F\subset\mathbb{R}^{128\times32\times32}$	2D feature space
$H\subset\mathbb{R}^{21\times32\times32}$	2D heatmap space
$M\subset\mathbb{R}^{778\times1538}$	3D mesh space: 778 vertices × 1,538 faces
$T\subset\mathbb{R}^{1538\times3}$	RGB mesh texture (3 × 1,538 faces)
$g^{\mathrm{FPE}}:X\to F\times H$	2D feature and pose estimator
$g^{\mathrm{HME}}:F\times H\to M$	Hand mesh estimator
$g^{\mathrm{Tex}}:F\times H\to T$	Texture estimator
$g^{\mathrm{NR}}:M\times T\to X$	Neural renderer [25]
$g^{\mathrm{Reg}}:M\to Y$	Hand joint regressor [54]
$g^{\mathrm{MR}}:F\times H\to X\times Y$	Mesh renderer: $g^{\mathrm{MR}}=[g^{\mathrm{NR}}\circ[g^{\mathrm{HME}},g^{\mathrm{Tex}}],g^{\mathrm{Reg}}\circ g^{\mathrm{HME}}]$
$g^{\mathrm{GAN}}:F\times H\times F\times H\to X$	GAN generator
$d^{\mathrm{GAN}}_1,d^{\mathrm{GAN}}_2:X\to\mathbb{R}$	GAN discriminators
$f^{\mathrm{DAN}}:X\to X\times Y$	Domain adaptation network: $f^{\mathrm{DAN}}=g^{\mathrm{MR}}\circ g^{\mathrm{FPE}}$



CPM Architecture

$g^{FPE}: X \to F \times H$ (2D maps $\{f, h\}$) $\to g^{MR}: F \times H \to X \times Y$ (x', y) $\to g^{GAN}: F \times H \times F \times H \to X$

f: 256^2 RGB (HOI) 128×32^2 spatial feature maps; h: 21×32^2 heatmaps

components: ① $\{f, h\} \to g^{HME}$ parallel g^{Tex}; ② $\{m, t\} \to g^{MR}$; ③ $m \to g^{Reg}$

g^{HME}: 63-dim MANO parameter p → MANO mesh m (proxy geometric features)

g^{Tex}: calculates mesh color t

g^{MR}: project m and t onto image plane, generate hand-only x'

g^{Reg}: determine 3D skeleton joints y

$\{x, x'\}$ → extract $\{f, h, f', h'\}$

Augment ✗ initial reconstruction refines search space of g^{GAN} parameters

$L_{img}(g^{FPE}, g^{HME}, g^{Tex} | D) = \frac{1}{2}\mathbb{E}[\log(1 - d_i^{GAN}(x''))] + \mathbb{E}[\log(1 - d_i^{GAN}(x'))] + \|x'' - x \odot S_{Hand}\|_1 + \|x' - x \odot S_{Hand}\|$

$L_{Heat}([g^{FPE}]_H | D_{Hand}) = \|g^{FPE}(x) - h_{GT}\|_2^2$

x', x'': Applied each iteration

$L_{pose}(g^{FPE}, g^{HME} | D_{Hand}) = \|[g^{MR}(g^{FPE}(x))]_Y - y_{GT}\|_2^2$