scientific reports

Check for updates

OPEN

Accurate 3D hand mesh recovery from a single RGB image

Akila Pemasiri✉, Kien Nguyen, Sridha Sridharan & Clinton Fookes

This work addresses hand mesh recovery from a single RGB image. In contrast to most of the existing approaches where parametric hand models are employed as the prior, we show that the hand mesh can be learned directly from the input image. We propose a new type of GAN called Im2Mesh GAN to learn the mesh through end-to-end adversarial training. By interpreting the mesh as a graph, our model is able to capture the topological relationship among the mesh vertices. We also introduce a 3D surface descriptor into the GAN architecture to further capture the associated 3D features. We conduct experiments with the proposed Im2Mesh GAN architecture in two settings: one where we can reap the benefits of coupled groundtruth data availability of the images and the corresponding meshes; and the other which combats the more challenging problem of mesh estimation without the corresponding groundtruth. Through extensive evaluations we demonstrate that even without using any hand priors the proposed method performs on par or better than the state-of-the-art.

Compared with existing 2D or 3D hand pose estimation from RGB or/and depth image data, hand mesh recovery can provide a more expressive and useful representation for monocular hand image understanding. The hand mesh recovery from a single RGB image is of particular interest for a wide range of applications in many domains, including augmented reality[1,2] and human computer interaction[3,4].

Hand mesh recovery is a challenging and ill-posed problem considering multiple meshes can be inferred from the same RGB image. The popular solution to deal with this ill-posed recovery is using priors. Most of the hand mesh recovery approaches in the literature employ the parametric MANO hand model[5] as the hand prior and employ some forms of neural networks to regress the model parameters[6–9]. However low dimensional nature of the parametric models limits their capability to capture non-linear shapes of hands[10]. In addition, some approaches rely on the heatmaps of the keypoint annotations in the early steps of their model[8,10]. We argue that this is redundant since the 3D keypoints can be learned simultaneously with the mesh and they should be learned simultaneously due to the complementary nature of two tasks.

In this paper, we approach the problem of learning the priors by end-to-end adversarial training. We show that the hand priors can be learned explicitly in the 3D mesh representation and can be encoded in a generative network. We propose a new type of Generative Adversarial Network (GAN) called Im2Mesh to learn the mesh vertices directly from a single RGB input image. Through the competing process of the generator and discriminator, the generator gradually improves to a level where it can generate the mesh directly from a single input image, providing an accurate solution for the hand mesh recovery task.

Importantly, by interpreting the mesh as a graph, we can employ recent advances in Graph Neural Networks (GNNs) to support mesh processing in both generator and discriminator networks. GNNs have demonstrated the capability of handling non-Euclidean structured data such as graphs and manifolds[11,12]. In contrast to the existing graph-based mesh estimation methods in the literature[10] which only consider the CNN generated features, we introduce a 3D descriptor that encodes surface level information into the GNNs, allowing them to better exploit the topological relationship among mesh vertices in the graph-structured hand data. This improves the mesh recovery accuracy since the recovery algorithm not only considers the vertex 3D coordinates but also the 3D features associated with the vertices.

Our main contribution of this paper are summarised as below:

- We propose a new GAN architecture called Im2Mesh to enable end-to-end learning of the hand mesh directly from a single RGB input image, without requiring any heatmap processing, 3D keypoint (joint landmark) annotations or external parametric hand models.
- We model the generator of the GAN as a graph architecture, allowing it to model the topological relationship among the vertices of the mesh. To the best of our knowledge, this is the first attempt to introduce a 3D

Signal Processing, Artificial Intelligence and Vision Technologies (SAIVT) Lab, Queensland University of Technology, 2 George Street, GPO Box 2434, Brisbane, QLD 4000, Australia. ✉email: akila@cse.mrt.ac.lk

descriptor into a generative graph model to encode the surface level information, explicitly capturing the 3D features associated with the mesh vertices.

- The proposed approaches not only address the problem of mesh reconstruction for the coupled datasets where one-to-one mapping prevails between the images and the groundtruth meshes, but also simultaneously address the problem of reconstructing meshes for the datasets which do not have the corresponding groundtruth annotations.
- We do not use the depth images; as such we increase the potential of using our model for the datasets which do not have the corresponding depth images.

The remainder of this paper is organised as follows. In section "Evaluations" we discuss the related work in the research area and their limitations. section "Conclusion" describes our methodology with subsections outlining each component where we describe the approach for the coupled data as well as for uncoupled data. In section 5 we present our experimental results including ablative studies and comparison of our method to the state-of-the-art, and we conclude the paper in section 5 while describing the results of the proposed methodology.

Related work

3D hand pose and mesh estimation using parametric models: Majority of existing 3D hand pose and shape estimation methods[8,13] are based on MANO[5] model which is a low dimensional parametric representation of hand mesh. However, there remain few weaknesses in using such parametric models. Firstly, the model is generated in controlled environments which are different from the images that are encountered in real world[14] thus causing a domain gap. Secondly, the low dimensional nature of the parametric models limits their capability to capture non-linear shapes of hands[10]. Thirdly, to create a parametric model it requires a large amount of data, which makes it challenging to adopt those methods to other object classes. Due to these limitations, in this paper we propose a hand mesh reconstruction approach which does not utilize a parametric hand model.

Model free 3D hand pose and mesh estimation: In the recent approaches on 3D hand pose and mesh estimation where parametric models are not used, other priors are employed. For an instance using 2D pose of the hand as an input to the network[14]. This requires the annotation of the 2D pose on input images, which limits the approach's ability in adopting to the datasets where the 2D pose annotation is not available. In addition, there exists some approaches that rely on heatmaps of the keypoints at early stages[8,10], which requires additional steps of keypoint estimation which can later be extracted directly from the estimated mesh. In contrast, we do not employ 2D or 3D keypoint locations in our method.

Graph Neural Networks (GNNs) for hand mesh estimation: In the recent literature, several approaches can be found where GNNs have been employed in estimating the 3D mesh of human hand[10,14]. However, the objective functions of these methods are limited to the vertex coordinates and other properties associated with the vertex location in the final mesh, where the resultant features of the GCNs are not fully utilized. To fully harness the strengths of GCNs we incorporate a 3D feature descriptor to our method, where the GCN is aimed to learn not only the vertex locations but it also learn to estimate the 3D feature descriptor, which elevate the overall accuracy of mesh estimation.

Effective use of datasets for hand pose and mesh estimation: When the datasets for hand pose and mesh estimation are considered, most recent datasets (Dome dataset by Kulon *et al.*[15], FreiHAND dataset by Zimmermann *et al.*[16]) contain the images and their corresponding groundtruth mesh. These datasets have been used by the state-of-the-art methods for hand pose and mesh estimation[13,14]. However, datasets such as RHD[17] and STB[18] contains the images and their corresponding groundtruth 3D pose, and the methods that have used those datasets have targeted only on estimating the 3D pose[19,20]. However, we propose an approach where the existing datasets which do not contain the groundtruth mesh details can effectively be used for the task of hand mesh estimation.

Im2Mesh GAN-single image mesh generation

When considering the available datasets for single image hand mesh reconstruction, there are 2 main variations; (1) The datasets which contain images and the corresponding by groundtruth mesh (i.e., the dataset by Kulon *et al.*[15] (referred as the Dome dataset hereafter) and the dataset by Zimmermann *et al.*[16] (referred as the FreiHAND dataset hereafter), and (2) Other standard datasets such as Rendered Handpose Dataset (RHD)[17] and Stereo Handpose Dataset (STB)[18] that do not contain the groundtruth meshes, instead they contain the 3D and 2D keypoint annotations of the human hand. Therefore we use two different network architectures: (1) To reap the maximum benefit of the availability of the coupled data in the Dome dataset[15] and FreiHAND dataset[16] and (2) To use the mesh data in Dome dataset along with the image data in other standard datasets (i.e., STB and RHD) for the robust estimation of the hand mesh.

In this section, we describe the details of our method. First, we briefly introduce the network architectures while distinguishing the architectural differences between the considered 2 settings, then we introduce the hand mesh representation and the 3D surface feature descriptor we use in this paper. We then elaborate on the details of each network architecture along with the objective functions that were employed.

Architecture for Coupled Data vs Architecture for the Uncoupled Data. The network architecture for the coupled training data is depicted in Fig. 1 and the network architecture for the uncoupled training data is depicted in Fig. 2. For the coupled training data a Conditional GAN architecture is employed (Eq. 3), where RGB images (*I*) are fed as the input to the "Generator". At the training time RGB images (*I*) with the generated meshes (*G*(*I*)) which contain the hand mesh representation (detailed in section "Evaluation protocol") and the 3D surface descriptor (detailed in section "Ablative studies") and RGB images with corresponding groundtruth meshes are fed to the "Discriminator".
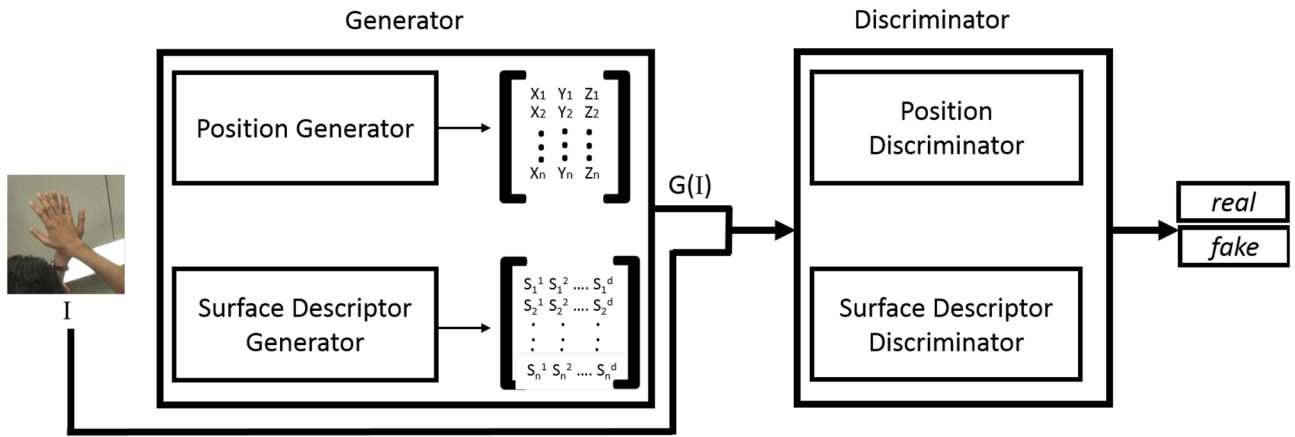
Figure 1. The proposed Im2Mesh GAN architecture for coupled training data. The position values and the SHOT descriptor values are generated using the generator network and passed to the discriminator network, which classifies whether they are generated or the groundtruth.
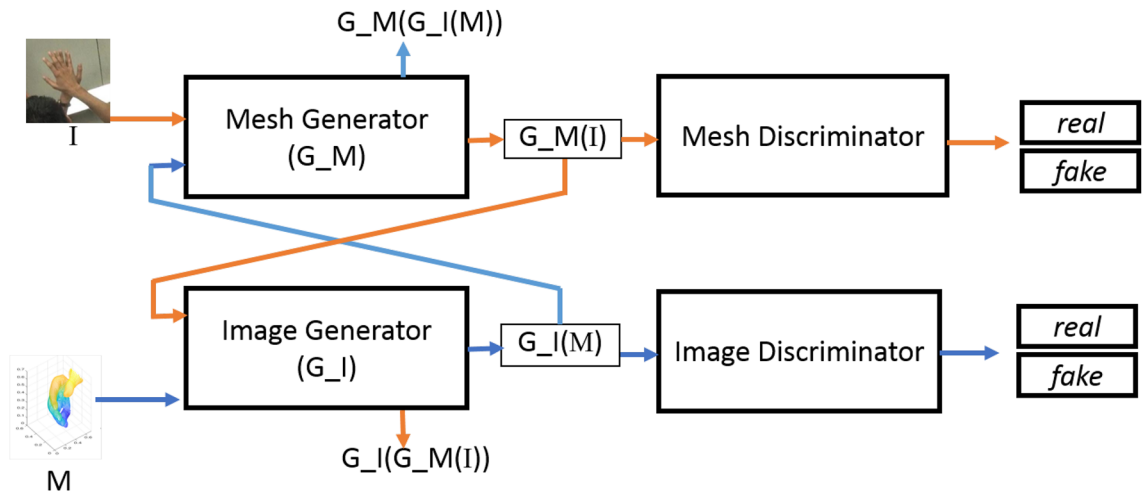

Figure 2. The proposed Im2Mesh GAN architecture for uncoupled training data. G_M denotes the generator which is designed for estimating the mesh from the input image I, where as G_I denotes the generator which is designed for estimating the image from the input mesh M.

For the uncoupled training data a Cycle GAN architecture is employed (Eq. 12) where RGB images from a particular dataset (e.g., RHD[17] dataset) is fed to the "Mesh Generator" (G_M) and mesh data from a different distribution (e.g. Dome dataset[16]) is fed to the "Image Generator" (G_I).

Hand mesh representation. In this work we represent hand mesh M as in Eq. (1), where V denotes the vertices and F denotes the faces that comprises the mesh. Each vertex in V is denoted by its x, y and z coordinates (i.e., $v_i = [x_i, y_i, z_i]$) and each face is denoted by the vertex numbers which have contributed for that face (i.e., $f_i = [v_p, v_q, v_r]$).

$$M = (V, F); V \, \varepsilon \, \mathbb{R}^{N \times 3}; F \, \varepsilon \, V^{M \times 3} \tag{1}$$

3D surface descriptor. In GNNs an attributed graph is defined as,

$$Graph = (V, E, X), \tag{2}$$

where V is the set of vertices/nodes which is directly extracted from M (Eq. 1), E is the set of edges which is derived using F in Eq. (1). X, can either be node attributes (i.e., $X^v \varepsilon \mathbb{R}^{N \times d}$ such that $X_{v_i} \varepsilon \mathbb{R}^d$, is the feature vector of node v_i), or edge attributes (i.e., $X^e \varepsilon \mathbb{E}^{T \times c}$ where T is the number of edges in the graph).

In this work, we use a node feature that can represent distinctive node properties. We selected the Signature of Histogram of Orientations (SHOT) descriptor[21], which has the ability to generate descriptive features for 3D points. SHOT descriptor is a combination of the concepts of "signature"[22] and "histogram"[23], such that the descriptor possess computational efficiency while maintaining the robustness. Apart from the evaluations that

have been performed by the developers of the SHOT descriptor Samuele *et al.*[21], the SHOT descriptor has demonstrated optimum performance in different domains[24], including in frameworks with deep learning techniques[25]. Furthermore compared with other 3D feature descriptors (e.g.Point Feature Histogram (PFH) descriptor[26] and Fast Point Feature Histogram (FPFH) descriptor[27]), SHOT descriptor the SHOT descriptor has been shown to better capture information of the surface as it encodes the details across radial, azimuth and elevation axes of the support region. The dimension of the feature descriptor depends on the parameters such as the number of neighbours that should be considered at the time of the feature descriptor creation.

Network architecture for coupled training data.   We use a variation of a conditional GAN in this work to generate realistic hand meshes, based on the RGB hand images. The objective of a conditional GAN is expressed as in Eq. (3), where G and D are the functions learned by the generator and the discriminator respectively. Conditional GANs are capable of learning the mapping between the input and the desired output.

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}\big[logD(x, y)\big] + \mathbb{E}_{x,z}\big[log(1 - D(x, G(x, z)))\big] \tag{3}$$

The configuration of the conditional GAN that we use is depicted in Fig. 1. The generator network has 2 components to predict the position vectors (i.e., V in Eq. 1) and the node features (i.e., X in Eq. 2, wherein this work we have used SHOT descriptor). Similarly, the discriminator network is also composed of 2 components, as "Position Discriminator" and "Surface Descriptor Discriminator".

When it comes to conditional GANs, the generator's objective is to generate output that closely resembles the ground-truth output while fooling the discriminator. Therefore, we define $\mathcal{L}(G)$ as in Eq. (4),

$$\mathcal{L}(G) = \lambda \mathcal{L}_{pos}(G) + \mu \mathcal{L}_{shot}(G) + \theta \mathcal{L}_{normal}(G) + \gamma \mathcal{L}_{Laplacian}(G) + \phi \mathcal{L}_{Quadratic}(G), \tag{4}$$

to measure the similarity between the predicted values and the corresponding groundtruth values.

The final objective of the conditional GAN is,

$$G^* = arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \delta \mathcal{L}(G) \tag{5}$$

The first two terms of Eq. (4) are aimed at minimizing the reconstruction error between the position vector and the SHOT descriptor respectively. $\mathcal{L}_{pos}$ is defined as,

$$\mathcal{L}_{pos} = \sum_{i=1}^{N} \left\| pred_{pos}^i - gt_{pos}^i \right\|_1, \tag{6}$$

where $pred_{pos}$ and gt_{pos} are the predicted and groundtruth vertex locations (i.e., position values) of the mesh[28].

We introduce $\mathcal{L}_{shot}$, which is the difference between the surface descriptors (SHOT descriptor) of the groundtruth mesh and the predicted mesh. $\mathcal{L}_{shot}$ is defined as in Eq. (7).

$$\mathcal{L}_{shot} = \sum_{i=1}^{N} \left\| pred_{shot}^i - gt_{shot}^i \right\|_1 \tag{7}$$

A loss based on the surface normals of the mesh is introduced to enforce the smoothness of the mesh. To ensure that the surface normals of the groundruth mesh and the predicted mesh are parallel, the dot product among them is used. $\mathcal{L}_{normal}$ is calculated as in Eq. (8), where n_{pred}^i and n_{gt}^i denote the normal vector of face i in the predicted mesh and the groundtruth mesh respectively.

$$\mathcal{L}_{normal} = \sum_{i=1}^{M} \left\| < n_{pred}^i, n_{gt}^i > \right\|_2^2 \tag{8}$$

In addition, to further enhance the smoothness of the mesh we employ the Laplacian loss ($\mathcal{L}_{Laplacian}$)[10]. We introduce two components to the Laplacian loss (Eq. 9), where the $\mathcal{L}_{VertexLaplacian}$ is calculated for each of the vertices in the mesh considering the adjacent neighbours while enforcing the smoothness in a fine grained context, and $\mathcal{L}_{KeypointLaplacian}$ is calculated for the keypoints while considering the neighbours in a broad range, thus enforcing the smoothness in a more coarser level. The weights of the $\mathcal{L}_{VertexLaplacian}$ and $\mathcal{L}_{KeypointLaplacian}$ are denoted by α and β. Laplacian error in general is defined as in Eq. (10), where $w_i = pred\_pos_i - gt\_pos_i$ for v_i. When considering $\mathcal{L}_{VertexLaplacian}$, the neighbours of vertex v_i are defined as $\mathcal{N}(v_i) = \{w \varepsilon V | (v_i, w) \varepsilon E\}$, whereas $\mathcal{L}_{KeypointLaplacian}$ is defined based neighbours that are identified through a graph unrolling and graph traversing process.

$$\mathcal{L}_{Laplacian} = \alpha \mathcal{L}_{VertexLaplacian} + \beta \mathcal{L}_{KeypointLaplacian} \tag{9}$$

$$Laplacian\ error = \sum_{i=1}^{N} \left\| \omega_i - \sum_{v_k \varepsilon \mathcal{N}(v_i)} \omega_k \Big/ B_i \right\|_2^2 \tag{10}$$

To define neighbours for $\mathscr{L}_{KeypointLaplacian}$ calculation, we unrolled the hand mesh into a graph format. For each of the keypoints, a separate graph is created by traversing the mesh using the vertex related to the keypoint as the starting node. We use breadth first search[29] based graph unrolling. As the vertices are not uniformly distributed throughout the mesh, the number of layers in each graph and the number of nodes in each layer is different.

The Quadratic loss ($\mathscr{L}_{Quadratic}$) (Eq. 11)[30] is also used to penalize the predicted points in the normal direction. In Eq. (11), $Q_{v_{gt}}v_{pred}$ stands for the quadratic error ([31,32]) which is calculated based on the triangle incidents that correspond to v_{gt}.

$$\mathscr{L}_{Quadratic} = \frac{1}{N}\sum_{\substack{v_{pred}\varepsilon M_{pred} \\ v_{gt}\varepsilon M_{gt}}} Q_{v_{gt}}v_{pred} \tag{11}$$

In general, the objective of the the discriminator network (D in Eq. 3) is to classify whether the given input is from the real sample or whether it has been generated by the generator network (G). However, the existing work on GANs is focused on discriminating the generated data such as class labels and images and thus have used fully connected or convolutional layers in the discriminator.

In this work we use graph convolutional layers in the "Surface Descriptor Discriminator" network (Fig. 1), where the node features are taken into consideration. As the edge connections (E in Eq. 2) remain the same for all the estimated meshes we use spectral based graph convolution operations. We used GCN layers introduced by Kipf et al.[33].

Network architecture for uncoupled training data.
Hand mesh estimation from a single image suffers from the problem of having a limited amount of training data and hence the deep learning based techniques with supervised learning can not be used. As a solution many of the existing methods use the datasets with 3D keypoint annotations and estimate the hand pose. The estimated pose is then used along with parametric models such as MANO[5] for the 3D mesh reconstruction.

In this paper, we use a variation of cycle GAN[34] to estimate the 3D mesh of hand using a single image, based on uncoupled training data. The overview of the framework that we use in this work is depicted in Fig. 2, where "Mesh Generator" and "Mesh Discriminator" consists of position and surface descriptor related components which are denoted in Fig. 1.

We define the objective function of the network as,

$$\mathscr{L}(G\_M, G\_I, D\_M, D\_I) = \mathscr{L}_{GAN}(G\_M, D\_M, I, M) + \mathscr{L}_{GAN}(G\_I, D\_I, M, I) + \delta\mathscr{L}_{cyc}(G\_M, G\_I), \tag{12}$$

where $\mathscr{L}_{GAN}$ is defined according to the adversarial loss proposed in[35] (Eq. 13). $\mathscr{L}_{cyc}$, which stands for cycle consistency loss is used to constraint the possible mapping functions such that the mapping from image to mesh can be made as unique as possible. We define the cycle consistency loss with two components as $\mathscr{L}_{cyc\_mesh}$ and $\mathscr{L}_{cyc\_im}$, which stand for the cycle consistency of the mesh generator and the image generator respectively. $\mathscr{L}_{cyc\_mesh}$ is defined as in Eq. (4), where we aim to retain the surface smoothness of the mesh while minimizing the position error and the shot descriptor error. It should be noted that in this setting $pred$ in Eqs. (6)–(9) refers to the $G\_M(G\_I(M))$ of Fig. 2. $\mathscr{L}_{cyc\_im}$ is defined as in Eq. (14).

$$\mathscr{L}_{GAN}(G, D, X, Y) = \mathbb{E}_y\left(logD(y) + \mathbb{E}_x log(1 - D(G(x)))\right) \tag{13}$$

$$\mathscr{L}_{cyc\_im} = \mathbb{E}\|G\_I(G\_M(I)) - I\|_1 \tag{14}$$

Generator and discriminator for coupled training data.
The Dome dataset[15], contains meshes with 7907 vertices and the FreiHAND dataset, which is based on the MANO model has 778 vertices. The generator network we use is composed with 2 main components 1) To estimate an initial mesh with low resolution and 2) To increase the mesh resolution. For the initial low resolution mesh, we targeted on learning position vectors and SHOT descriptors for 224 vertices. To be compatible with the image shape we derived the shot descriptors with a dimension of 221. For this dataset, SHOT descriptors of a dimension of 221 were obtained by setting the parameters such that *number of bins* = 7, *radius of descriptor estimation* = 3 and *minimum neighbours* = 3.

Generator

For the preliminary layers of the generator we used a convolution based architecture which is in the shape of a U-Net[36], however we did not use any skip connections in this work as the domains of the input and output are different. In denoting the 2D convolution layers followed by a batch normalization and a ReLU activation we use the notation of Convolution-BatchNorm-ReLU. We used a similar architecture as in[37], where the encoder has 8 layers of Convolution-BatchNorm-ReLUs with 64, 128, 256, 512, 512, 512, 512 and 512 kernels followed by a decoder with 8 layers of Convolution-BatchNorm-ReLUs with 512, 512, 512, 512, 256, 128, 64. The above mentioned convolutions are 4×4 filters and decoder network is followed by another convolution layer to make the output channel dimension to 1. The ReLUs in the encoder are leaky with a slope of 0.2. After the final pass, which results in an output of shape 224×224 (ignoring the batchsize dimension and channel dimension), we decompose that into two components as [224, 3] and [224, 221], where the first component is the position vector and the second component is the SHOT descriptor for the 224 vertices in the coarse grained mesh.
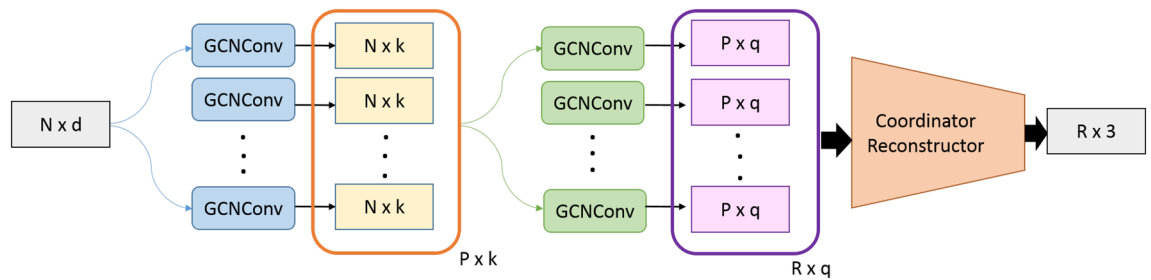
Mesh enhancer

Figure 3. The mesh enhancement process. It should be noted that this image depicts the process of upsampling a graph which has N nodes and a feature dimension of d, to a graph with R nodes. The depicted network contains two cascaded levels of graph upsampling followed by the "Coordinate Reconstructor" which calculates the position vector of the upsampled graph. k and q are the feature dimensions that of the generated features at cascaded level 1 and 2. Since the objective of our work is to upsample the graph while retaining the number of features we set $k = q = d$.

To enhance the mesh resolution, we used a cascade of Multi-branch GCN[38] modules, where GCNConv layers[33] were used for feature upsampling. For the Dome dataset we used 5 Multi-branch GCN modules at the first cascade level and then 8 modules in the second cascaded level and for the FreiHAND dataset we use 3 Multi-branch GCN modules. The resultant node features which construct mesh at full resolution were then passed through a set of Convolution-BatchNorm-ReLU which plays a role analogues to the role of "Coordinator Reconstructor" of the initial work on point upsampling[38]. This contains Convolution-BatchNorm-ReLU layers with 64, 64, 64, 64 and 1 kernels with each kernel having 1×3 filters. Thus the output of this network constructs the position vector for the mesh at full resolution.

Discriminator

The discriminator network contains 2 branches where the position vectors and the node features are compared with the corresponding groundtruth values. "Position Discriminator" takes the input of size $N \times 3$ and the "Surface Descriptor Discriminator" takes the input of size $N \times 221$, where N is the number of vertices in the resultant mesh. For the position discriminator we use three Convolution-BatchNorm-ReLU layers with 64,64 and 1 kernels in each with 5×1 filters and for the "Surface Descriptor Discriminator", we use 2 GCNConv layers where we reduce the feature size from 221 to 100 and then to 50. We pass each of these through a fully connected layer with 2048 nodes and then concatenate the output from the 2 discriminators before passing them trough another fully connected layer of 1048 nodes followed by a fully connected layer with size 1 and softmax activation.

Generator and discriminator for uncoupled training data. Compared with the network settings that are described in the above section where the coupled training data is available, when our implementation of cycle GAN (Fig. 2) is considered the main difference is that in cycle GAN we have enforced the cycle consistency loss where we feed the mesh generated by "Mesh Generator" in Fig. 2 to the "Image Generator" and vice-versa. To allow this, the inputs for each generator should be same in shape. Hence we consider the coarse grained mesh which has 224 vertices and set the input image width and height to 224 pixels. For the generators and for the "Image Discriminator" we used the same architectures that have been used by the original paper of cycle GAN[34]. For the mesh discriminator, we used the same architecture that we described in the "Discriminator" subsection of section "3.6".

For the setting of uncoupled training data, we separately trained the mesh enhancer (Fig. 3) such that it learns the mapping from low dimensional mesh to the high dimensional mesh, and the generated low dimensional meshes were upsampled using it.

Training. For the network which uses coupled data we set $\delta = 10$ in Eq. (5). All the parameters $\lambda, \mu, \theta, \gamma$ and *phi* in Eq. (4), and α and β in Eq. (9) were set to 1. To see the effectiveness of the constraints that were enforced to obtain surface smoothness we evaluate the method with setting the parameters θ and γ to 0. The results related to those settings can be found in the section "Conclusion". For the network which used uncoupled data, δ was set to 10. We used a training procedure similar to[34,37] where all the models were trained from the scratch and with a learning rate of 0.0002 using Adam optimizer[39].

Evaluations

In this section we describe the datasets and the evaluation matrix that we used, the ablative studies that we conducted to evaluate the effectiveness of the components in our model, the experimental results that we obtained in benchmarking our model with the state-of-the-art methods. It should be noted that when recording the results of the state-of-the-art methods, we have used the evaluations that are been performed by the respective authors and the results with the best configuration of their proposed methods have been selected for the comparison.

Datasets. This work utilizes two types of publicly available datasets, (1) Coupled dataset where the images and the corresponding groundtruth mesh is available and (2) Datasets which contain only the images. For the coupled dataset we used the Dome dataset[15] and FreiHAND dataset[16] and for the later we used the RHD[17] and

Use of loss components					
3D Surface descriptor loss	Surface normal loss	Laplacian loss		Quadratic Loss	Reconstruction error (mm)
		Vertex Laplacian	Keypoint Laplacian		
✗	✓	✓	✓	✓	2.62
✓	✗	✗	✗	✓	2.93
✓	✗	✓	✓	✓	2.85
✓	✓	✗	✗	✓	2.73
✓	✓	✗	✓	✓	2.70
✓	✓	✓	✗	✓	2.67
✓	✓	✓	✓	✗	1.81
✓	✓	✓	✓	✓	1.79

Table 1. Results of the conducted ablative studies to evaluate the effectiveness of the loss components that are indicated in Eq. (4) and in Eq. (9). The first row demonstrates the importance of using the 3D surface descriptor, where as the next five rows (highlighted in gray) demonstrate the effectiveness of surface smoothness and the seventh row demonstrates the effectiveness of the quadratic loss.
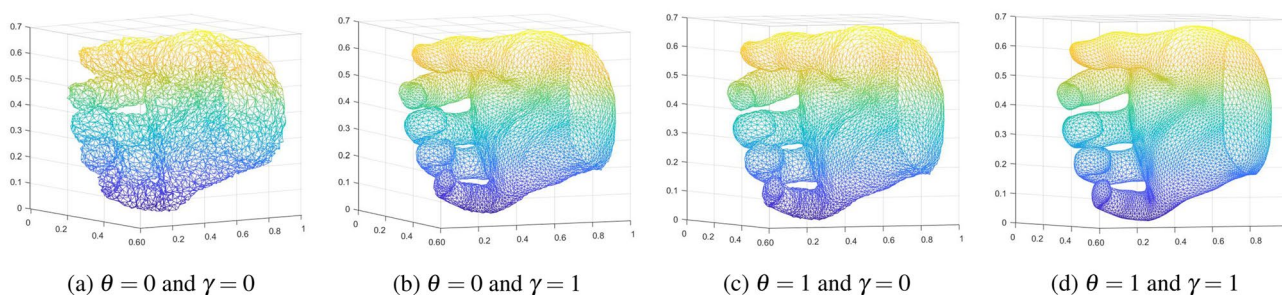

(a) $\theta = 0$ and $\gamma = 0$ (b) $\theta = 0$ and $\gamma = 1$ (c) $\theta = 1$ and $\gamma = 0$ (d) $\theta = 1$ and $\gamma = 1$

Figure 4. The qualitative results that were obtained by changing the parameters that are related to surface smoothness, in Eq. (4).

STB[18] hand datasets with the 3D meshes which are available as the groundtruth in Dome dataset. These datasets have been widely used for benchmarking hand shape and pose estimation[8,17,40–42].

Evaluation protocol. For the Dome dataset we used the L1 reconstruction error, between the ground truth mesh and the predicted mesh. For the quantitative evaluations of the datasets for which groundtruth meshes are not available we extracted the 3D locations of the keypoints and used the accuracy measurement of Percentage of Correct Keypoints (PCK) scores. In PCK calculation, if the predicted keypoint, which we extract from the estimated mesh lies within a sphere with a specific radius with respect to the groundtruth value, it is considered as correct keypoint.

Ablative studies. Our ablative studies were conducted on the Dome dataset which has the groundtruth mesh data, as such the quantitative evaluations could be carried out. We conducted ablative studies, with the aim of evaluating the contribution of the components in Eq. (4). The results related to the secon ablative studies are recorded in Table 1.

Effectiveness of using the 3D surface descriptor: We performed this ablative study to evaluate the effectiveness of incorporating $\mathcal{L}_{shot}$, which measures the similarity in the groundtruth and generated SHOT descriptor. We trained our model on training subset and tested on the test subset of the Dome dataset[15]. For this ablative study, we set $\mu = 0$ in Eq. (4).

Effectiveness of enforcing the surface smoothness in the mesh: As described in section "Selection of the 3D feature descriptor", our method combines several loss functions to enforce the surface smoothness of the mesh. We evaluate the effectiveness of each of these loss components (i.e., $\mathcal{L}_{normal}$ and $\mathcal{L}_{Laplacian}$, where the later consists of 2 components as $\mathcal{L}_{VertexLaplacian}$ and $\mathcal{L}_{KeypointLaplacian}$). We assessed the effectiveness of using components individually and in combination. First we eliminate all the loss values that are related to the surface smoothness (i.e., surface normal loss, vertex laplacian loss and the keypoint laplacian loss), and the corresponding results can be found in the second raw of Table 1. Similarly by eliminating individual lossless we compared the reconstruction error (Table 1). Figure 4 better visualizes the effect of constraints that are used to smooth the surface. From the conducted ablative studies it is evident that the contribution of the components in the loss function is vital and the compound of all the components has resulted in better accuracy values.

3D feature descriptor	Reconstruction Error (mm)
PFH descriptor[26]	4.31
FPFH descriptor[27]	3.87
SHOT descriptor[21]	1.79

Table 2. Experimental results of replacing SHOT with other 3D descriptors in the evaluation on the Dome dataset[15].

	Method	Pose estimation	Shape estimation	Parametric models are not required	AUC Value
Requires 2D joint locations	GE et al. (CVPR, 2019)[10]	✓	✓	✓	0.92
	Baek et al. (CVPR, 2019)[43]	✓	✓	✗	0.926
Does not require 2D joint locations	Zimmermann et al. (ICCV, 2017)[17]	✓	✗	✓	0.675
	Spurr et al. (CVPR, 2018)[40]	✓	✗	✓	0.849
	Cai et al. (ECCV, 2018)[41]	✓	✗	✓	0.887
	Zhang et al. (ICCV, 2019)[8]	✓	✓	✗	0.901
	Pemasiri et al. (BMVC, 2019)[42]	✓	✗	✓	0.911
	Yang et al. (ICCV, 2019)[44]	✓	✗	✓	0.943
	Kulon et al. (CVPR, 2020)[9]	✓	✓	✗	0.956
	Ours	✓	✓	✓	0.946

Table 3. Area under curve values for RHD dataset, which are related to Fig. 5a.

	Method	Pose estimation	Shape estimation	Parametric models are not required	AUC Value
Requires 2D joint locations	Muller (CVPR, 2018)[45]	✓	✗	✓	0.965
	Iqbal et al. (ECCV, 2018)[19]	✓	✗	✓	0.994
	Ge et al. (CVPR, 2019)[10]	✓	✓	✓	0.998
Does not require 2D joint locations	Sun et al. (CVPR, 2015)[46]	✓	✗	✓	0.839
	Zimmermann et al. (ICCV, 2017)[17]	✓	✗	✓	0.948
	Spurr et al. (CVPR, 2018)[40]	✓	✗	✓	0.983
	Cai et al. (ECCV, 2018)[41]	✓	✗	✓	0.993
	Panteleris et al. (WACV, 2018)[20]	✓	✗	✓	0.994
	Zhang et al. (ICCV, 2019)[8]	✓	✓	✗	0.995
	Ours	✓	✓	✓	0.998

Table 4. Area under curve values for STB dataset which are related to Fig. 5b.

Selection of the 3D feature descriptor. We performed a comparative study to confirm that the SHOT descriptor is the best choice when compared to other 3D surface descriptors. For this evaluation we used Point Feature Histogram (PFH) descriptor, Fast Point Feature Histogram (FPFH) descriptor and SHOT descriptor. We trained our model by using each of these descriptors in Eq. (4) on training subset and tested on the test subset of the Dome dataset[15]. The results are recorded in Table 2 and clearly demonstrate the superior performance of the SHOT descriptor (Tables 3, 4).

Comparison to the state-of-the-art. For the Dome dataset we compared our method with the state-of-the-art method which has used the same dataset. When the two methods were compared using the L1 reconstruction error, the obtained results are recorded in Table 5. From the obtained values it can be seen that our method has outperformed the state-of-the-art method with significant margins. Similarly for FreiHAND dataset, we compared our results with the state-of-the-art methods and the obtained results are recorded in Table 6.

For the RHD dataset and STB dataset, the evaluations were performed using the 3D PCK values, where we compare the extracted 3D keypoint values with the groundtruth 3D keypoint values. The obtained PCK values are recorded in Fig. 5. From the results it can be seen that, though the objective of our method was not estimating the 3D keypoint locations, with our method we have obtained PCK values which are comparable to the

Method	Reconstruction error (mm)
Kulon *et al.* (BMVC, 2019)[15]	2.33
Ours	1.83

Table 5. L1 Reconstruction error which was obtained for the Dome dataset[15].

	Method	Pose and shape estimation	PA MPVPE	F@5mm	F@15mm
Use parametric models	Hasson *et al.* (CVPR, 2019)[7]	✓	13.2	0.436	0.908
	Boukhayma *et al.* (CVPR, 2019)[6]	✓	13	0.435	0.898
	FreiHAND (ICCV, 2019)[16]	✓	10.7	0.529	0.935
	Pose2Mesh (ECCV, 2020)[14]	✓	7.8	0.674	0.969
	I2L-MeshNet (ECCV,2020)[13]	✓	7.6	0.681	0.973
Does not use parametric models	Im2Mesh GAN (Ours)	✓	7.6	0.681	0.973

Table 6. The PA MPVPE and F-scores which were obtained for the FreiHAND dataset[16].

(a) 3D PCK values on RHD dataset .

(b) 3D PCK values on STB dataset.
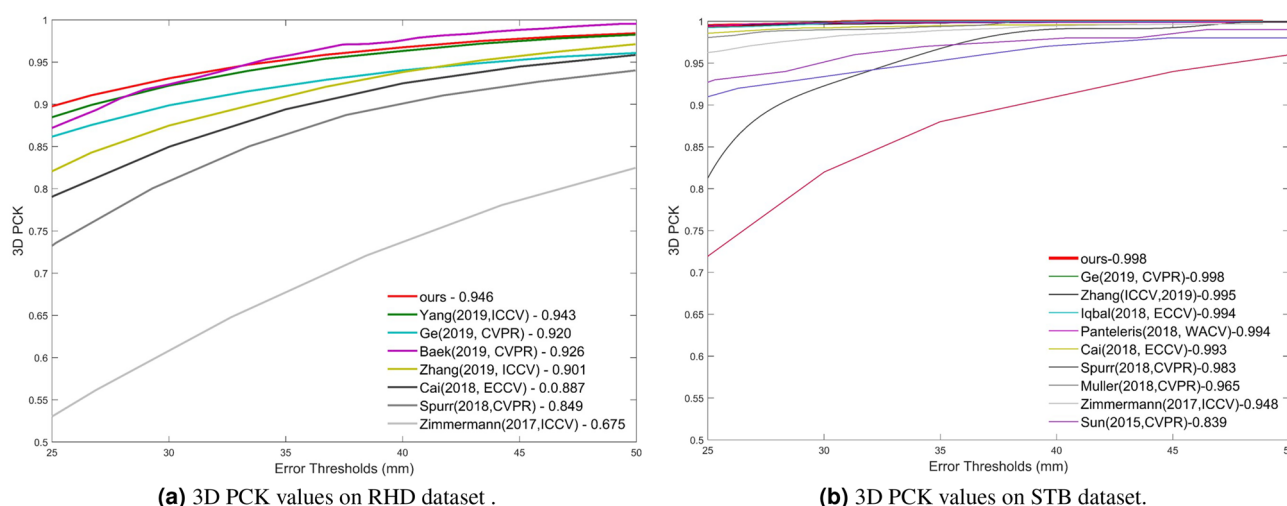
Figure 5. The PCK values that were obtained when comparing our method with the state-of-the-art methods, where our method has comparable results to the state-of-the-art methods.

state-of-the-art methods. It should be noted that our method has not been trained with 3D pose supervision. However, it outperforms the state-of-the-art methods that have been trained with 3D pose supervision. Furthermore to demonstrate the constrains of other methods we have highlighted their capabilities and limitations in Tables 3, 4 and 6. Area Under Curve (AUC) values related to Fig. 5a are recorded in Table 3 and the AUC values related to Fig. 5b are recorded in Table 4.

It should be noted that when comparing the quantitative performance of the state-of-the-art methods, we have used the evaluations that are been performed by respective authors and the results with the best configurations of their proposed methods have been considered.

The qualitative results that were obtained for the coupled datasets (i.e., Dome dataset[15] and FreiHand dataset[16]) are depicted in figure where the meshes are consisted with 7907 vertices are depicted in Figs. 6 and 7 respectively. It should be noted that the Dome dataset contains meshes with 7907 vertices while FreiHand[16] dataset contains the meshes with 778 vertices. In Fig. 7 the 5th column depicts the mask that was obtained by projecting the estimated mesh. Qualitave results for the RHD dataset[17], which is an uncoupled dataset are depicted in Fig. 8.
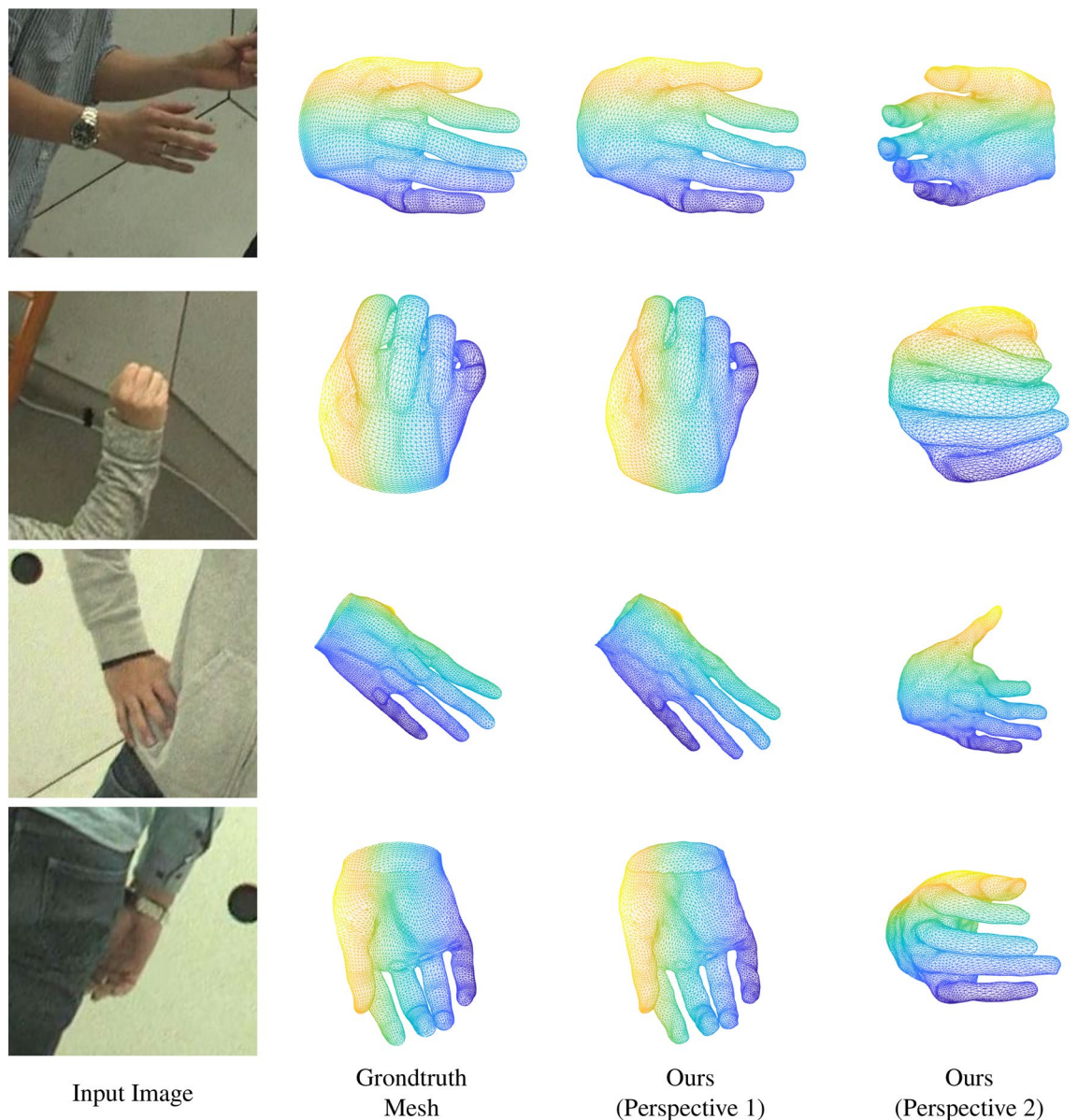
Figure 6. The qualitative results obtained for the Dome dataset[15]. The meshes in the dataset contain 7907 vertices.

Conclusion

While 3D mesh reconstruction of the human hand using a single image has been explored in the past, the problem still remains a challenge due to the high degree of freedom of the human hand. In this paper, we have presented a method to create 3D mesh of the hand using the single image that can effectively use the existing databases for better reconstruction of the 3D mesh using a single image. We have designed a loss function that can generate more realistic hand meshes, and we demonstrate the effectiveness of that loss function in two settings of Generative Adversarial Networks. The first setting is targeted on the effective use of coupled datasets where the groundtruth meshes are available, whereas the second setting is targeted on uncoupled datasets. In addition, we employ a 3D surface descriptor in this work along with graph convolution networks, which enable the preservation of the surface details of generated meshes. We confirm that our framework outperforms the state-of-the-art as well represents the first effort to incorporate explicit 3D features in a single image-based 3D mesh reconstruction. One of the interesting properties of the proposed mesh recovery approach is that there is no need for parametric hand models as priors. The geometry of the hand is learned and encoded directly in the generator through the end-to-end adversarial training process. This fact enables the proposed algorithm to be easily adapted to other mesh problems such as other body parts or 3D objects.
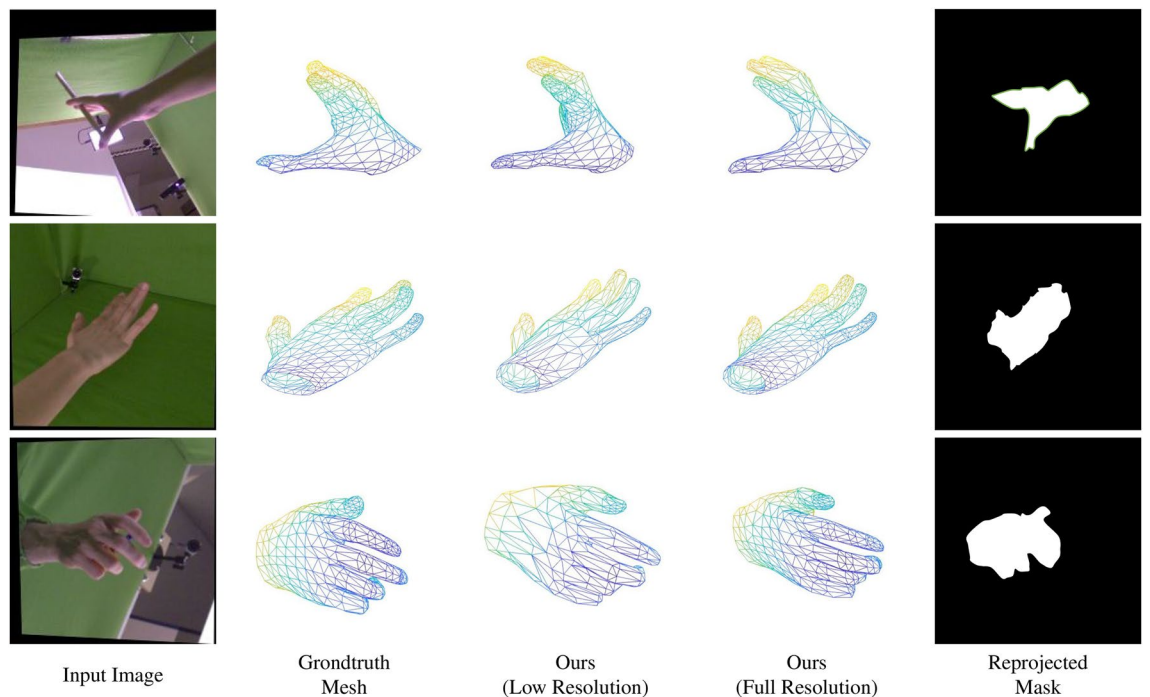
| Input Image | Grondtruth Mesh | Ours (Low Resolution) | Ours (Full Resolution) | Reprojected Mask |

Figure 7. The qualitative results obtained for the FreiHand dataset[16]. The meshes in the FreiHAND dataset is based on MANO model[5] and thus contains 778 vertices.
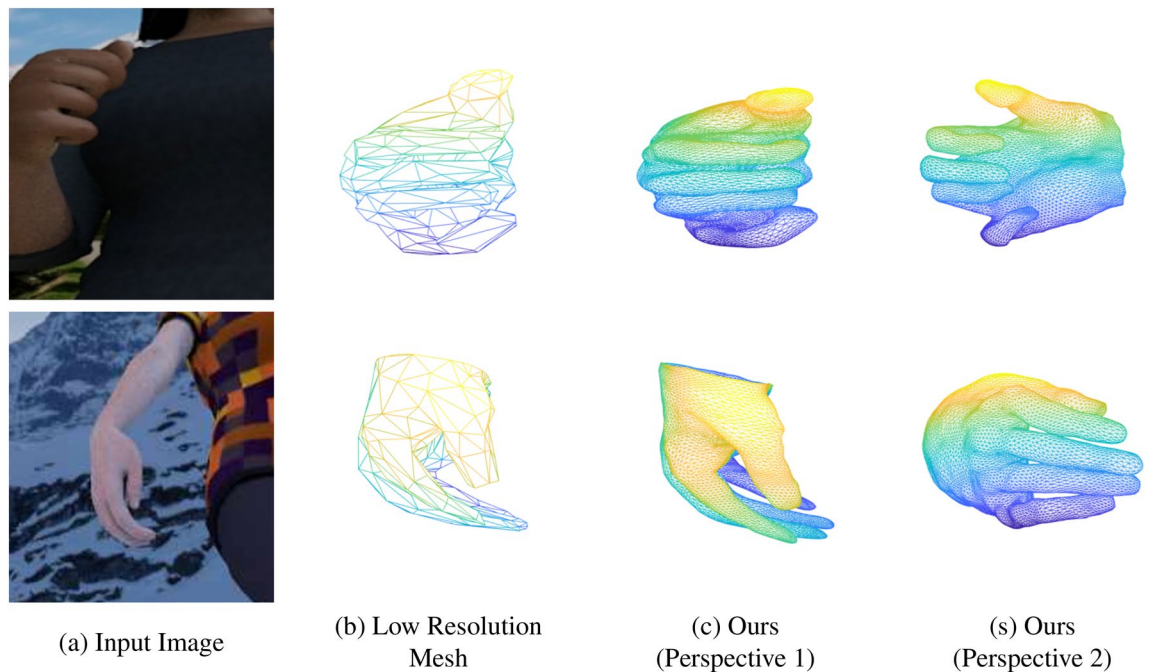

(a) Input Image   (b) Low Resolution Mesh   (c) Ours (Perspective 1)   (s) Ours (Perspective 2)

Figure 8. The qualitative results that were obtained for the model with uncoupled training data[17].

References

1. Lee, T. & Hollerer, T. Multithreaded hybrid feature tracking for markerless augmented reality. *IEEE Trans. Visual Comput. Graphics* **15**, 355–368 (2009).
2. Piumsomboon, T., Clark, A., Billinghurst, M. & Cockburn, A. User-defined gestures for augmented reality. In *IFIP Conference on Human–Computer Interaction*, 282–299 (Springer, 2013).

3. Markussen, A., Jakobsen, M. R. & Hornbæk, K. Vulture: a mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1073–1082 (2014).

4. Sridhar, S., Feit, A. M., Theobalt, C. & Oulasvirta, A. Investigating the dexterity of multi-finger input for mid-air text entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 3643–3652 (2015).

5. Romero, J., Tzionas, D. & Black, M. J. Embodied hands: modeling and capturing hands and bodies together. *ACM Trans. Graphics* **36**, 1–17 (2017).

6. Boukhayma, A., Bem, R. D. & Torr, P. H. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10843–10852 (2019).

7. Hasson, Y. *et al.* Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11807–11816 (2019).

8. Zhang, X., Li, Q., Mo, H., Zhang, W. & Zheng, W. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2354–2364 (2019).

9. Kulon, D., Guler, R. A., Kokkinos, I., Bronstein, M. M. & Zafeiriou, S. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4990–5000 (2020).

10. Ge, L. *et al.* 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10833–10842 (2019).

11. Monti, F. *et al.* Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5115–5124 (2017).

12. Wu, Z. *et al.* A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 4–24 (2020).

13. Moon, G. & Lee, K. M. I2l-meshnet: image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, 752–768 (Springer, 2020).

14. Choi, H., Moon, G. & Lee, K. M. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision*, 769–787 (Springer, 2020).

15. Kulon, D., Wang, H., Güler, R. A., Bronstein, M. & Zafeiriou, S. Single image 3d hand reconstruction with mesh convolutions. arXiv preprint arXiv:1905.01326 (2019).

16. Zimmermann, C. *et al.* Freihand: a dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 813–822 (2019).

17. Zimmermann, C. & Brox, T. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, 4903–4911 (2017).

18. Zhang, J. *et al.* A hand pose tracking benchmark from stereo matching. In *2017 IEEE International Conference on Image Processing (ICIP)*, 982–986 (IEEE, 2017).

19. Iqbal, U., Molchanov, P., Gall, T. B. J. & Kautz, J. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 118–134 (2018).

20. Panteleris, P., Oikonomidis, I. & Argyros, A. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 436–445 (IEEE, 2018).

21. Salti, S., Tombari, F. & Di Stefano, L. Shot: unique signatures of histograms for surface and texture description. *Comput. Vis. Image Underst.* **125**, 251–264 (2014).

22. Chua, C. S. & Jarvis, R. Point signatures: a new representation for 3d object recognition. *Int. J. Comput. Vis.* **25**, 63–85 (1997).

23. Johnson, A. E. & Hebert, M. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**, 433–449 (1999).

24. Guo, Y. *et al.* A comprehensive performance evaluation of 3d local feature descriptors. *Int. J. Comput. Vis.* **116**, 66–89 (2016).

25. Boscaini, D., Masci, J., Rodolà, E. & Bronstein, M. Learning shape correspondence with anisotropic convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **29** (2016).

26. Rusu, R. B., Blodow, N., Marton, Z. C. & Beetz, M. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3384–3391 (IEEE, 2008).

27. Rusu, R. B., Blodow, N. & Beetz, M. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE International Conference on Robotics and Automation*, 3212–3217 (IEEE, 2009).

28. Wang, N. *et al.* Pixel2mesh: generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 52–67 (2018).

29. Bundy, A. & Wallen, L. Breadth-first search. In *Catalogue of Artificial Intelligence Tools*, 13–13 (Springer, 1984).

30. Agarwal, N., Yoon, S.-E. & Gopi, M. Learning embedding of 3d models with quadric loss. arXiv preprint arXiv:1907.10250 (2019).

31. Garland, M. & Heckbert, P. S. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 209–216 (1997).

32. Ronfard, R. & Rossignac, J. Full-range approximation of triangulated polyhedra. In *Computer Graphics Forum*, vol. 15, 67–76 (Wiley Online Library, 1996).

33. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016).

34. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2223–2232 (2017).

35. Goodfellow, I. *et al.* Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **27** (2014).

36. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, 234–241 (Springer, 2015).

37. Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1125–1134 (2017).

38. Qian, G., Abualshour, A., Li, G., Thabet, A. & Ghanem, B. Pu-gcn: point cloud upsampling using graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11683–11692 (2021).

39. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

40. Spurr, A., Song, J., Park, S. & Hilliges, O. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 89–98 (2018).

41. Cai, Y., Ge, L., Cai, J. & Yuan, J. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 666–682 (2018).

42. Hewa Thondilege, A. S. P., Nguyen Thanh, K., Sridharan, S. & Fookes, C. Unified 2d and 3d hand pose estimation from a single visible or x-ray image. In *Proceedings of the 30th British Machine Vision Conference 2019, BMVC 2019* (British Machine Vision Association, 2019).

43. Baek, S., Kim, K. I. & Kim, T.-K. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1067–1076 (2019).

44. Yang, L. & Yao, A. Disentangling latent hands for image synthesis and pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9877–9886 (2019).

45. Mueller, F. *et al.* Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 49–59 (2018).

46. Sun, X., Wei, Y., Liang, S., Tang, X. & Sun, J. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 824–832 (2015).

Acknowledgements

Author contributions

A.P.: Conception of approach and design of experiments, conducting experiments, analysis of results and writing the original draft of the manuscript. K.N.: Problem formulation, conceptualisation, analysis of results, reviewing and editing the manuscript. S.S.: Problem formulation, conceptualisation, analysis of results, reviewing and editing the manuscript. C.F.: Problem formulation, conceptualisation, analysis of results, reviewing and editing the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.