

TSCom-Net: Coarse-to-Fine 3D Textured Shape Completion Network

Ahmet Serdar Karadeniz, Sk Aziz Ali, Anis Kacem, Elona Dupont, and
Djamila Aouada

SnT, University of Luxembourg, L-1511 Luxembourg
{firstname.lastname}@uni.lu

Abstract. Reconstructing 3D human body shapes from 3D partial textured scans remains a fundamental task for many computer vision and graphics applications – *e.g.*, body animation, and virtual dressing. We propose a new neural network architecture for 3D body shape and high-resolution texture completion – TSCom-Net – that can reconstruct the full geometry from mid-level to high-level partial input scans. We decompose the overall reconstruction task into two stages – first, a joint implicit learning network (SCom-Net and TCom-Net) that takes a voxelized scan and its occupancy grid as input to reconstruct the full body shape and predict vertex textures. Second, a high-resolution texture completion network, that utilizes the predicted coarse vertex textures to inpaint the missing parts of the partial ‘texture atlas’. A Thorough experimental evaluation on 3DBodyTex.V2 dataset shows that our method achieves competitive results with respect to the state-of-the-art while generalizing to different types and levels of partial shapes. The proposed method has also ranked second in the track1 of SHape Recovery from Partial textured 3D scans (SHARP [37,2] 2022¹ challenge1.

Keywords: 3D Reconstruction, Shape Completion, Texture-Inpainting, Implicit Function, Signed Distance Function

1 Introduction

3D textured body shape completion and reconstruction from visual sensors plays a key role in a wide range of applications such as gaming, fashion, and virtual reality [31,30]. The challenges of this task and the methods to tackle it varies depending on the used sensors to capture the human shape and texture. Some existing methods focus mainly on 3D textured body shape reconstruction from a single monocular image [19,42,22,17,49]. For instance, under the body shape symmetry assumption, methods like Human Mesh Recovery (HMR) [18], use a statistical body model [25] to reconstruct the body shape and pose. Despite the impressive advances in this line of work [42,22,49], the reconstructed shapes still cannot capture high-level geometrical details, such as wrinkles, ears. This is due to the lack of original geometrical information and the projective distortions in

¹ <https://cvi2.uni.lu/sharp2022/>

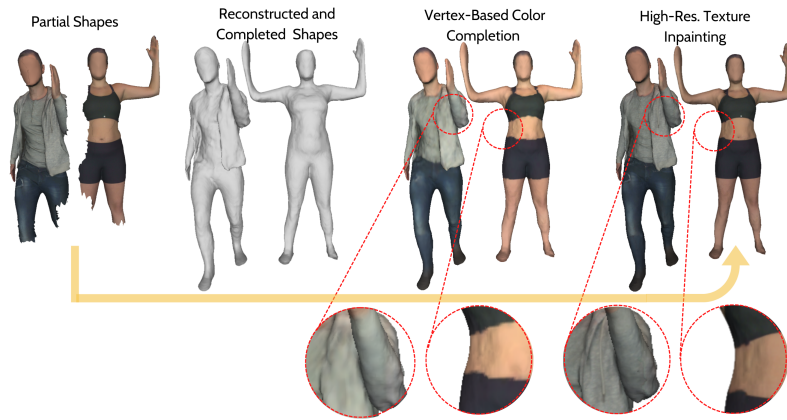


Fig. 1. 3D Partial Textured Body Shape Completion. The proposed deep learning-based method reconstructs and completes the surface geometry of a dressed or minimally-clothed partial body scan, and inpaints the missing regions with high-resolution texture. TSCom-Net has the flexibility to either apply dense texture mapping or vertex-based color estimation as per desired application scenarios.

2D images. On the other hand, 3D scanners and RGB-D sensors can provide richer information about the geometry of body shapes [41]. Nevertheless, they are often subject to partial acquisitions due to self occlusions, restricted sensing range, and other limitations of scanning systems [46,8]. While most of existing works focus on completing the geometry of body shapes from 3D partial scans [8,46,38], less interest has been dedicated to complete both the texture and the geometry at the same time. Nonetheless, this problem remains critical in real-world applications where complete and realistic human reconstructions are usually required. Aware of this need, recent competitions, such as SHAPe Recovery from Partial Textured 3D Scans (SHARP) challenges [3,1,37], emerged in the research community to foster the development of joint shape and texture completion techniques from partial 3D acquisitions. The results of these competitions showed promising techniques [9,36] with a room for improvements.

The problem of textured 3D body shape completion from 3D partial scans, as shown in Fig. 1, can naturally be decomposed into two challenging subtopics – (i) partial shape surface completion and (ii) partial texture completion of the reconstructed shape. In this setup, especially when no fixed UV-parametrization of shape is available, methods need to rely on inherent shape structure and texture correlation cues from the available body parts. Taking this direction, IFNet-Texture [9] uses implicit functions to perform high-quality shape completion and vertex-based texture predictions. Despite the impressive results in SHARP challenges [37,1], the method [9] does not output high-resolution texture and often over-smooths vertex colors over the whole shape. On the other hand, 3DBooSTeR method [36] decouples the problems of shape and texture completions and solves them in a sequential model. The shape completion of [36] is

based on body models [12], while the texture completion is tackled as an image based *texture-atlas* inpainting. [36] can complete high-resolution partial texture, but suffers from large shape and texture modelling artifacts.

Our method overcomes the weaknesses of both [9] and [36]. The shape completion part of our TSCom-Net is designed to learn a continuous *implicit function* representation [28,48,29] as also followed by Chibane *et al.* [8]. Furthermore, we identify that the IFNet-Texture method [9], an extension of [8], uses a vertex-based inference-time texture estimation process that ignores the shape and color feature correlation cues at the learning phase. For this reason, we employ an end-to-end trainable vertex-based shape and texture completion networks – SCom-Net and TCom-Net – to boost their performance with joint-learning. Next, we propose to refine the predicted vertex-based texture completions directly in the texture-atlas of 3D partial scans. This is achieved using an image inpainting network [23,36], reusing the predicted vertex textures, and yielding high-resolution texture. It is notable that, unlike [20,36], TSCom-Net does not require a fixed UV-parametrization which makes the considered ‘texture charts’ random, discontinuous, and more complex in nature across the samples. Overall, our **contributions** can be summarized to:

1. An end-to-end trainable joint implicit function network for 3D partial shape and vertex texture completion. We propose an early fusion scheme of shape and texture features within this network.
2. A high-resolution texture-atlas inpainting network, which uses partial convolutions [23] to refine the predicted vertex textures. At the same time, this module is flexible and can be plugged with any other 3D shape and vertex texture reconstruction module.
3. An extensive experimental evaluation of TSCom-Net showing its multi-stage improvements, its comparison *w.r.t.* the participants in SHARP 2022 challenge [3], and its generalization capabilities to different types of input data partiality.

The rest of the paper is organized as follows. After summarizing the methods related to our line of work in Section 2, we present the core components of TSCom-Net in Section 3. The experiments and evaluation parts are reported and discussed in Section 4. Finally, Section 5 concludes the paper and draws some future works.

2 Related Works

Deep Implicit Function and 3D Shape Representation Learning. Supervised learning methods for 3D shape processing require an efficient input data representation. Apart from common learning representations of 3D shapes – such as regular voxel grids [24,49], point clouds [32], Oc-trees [44], Barnes-Hut 2^D -tree [5], depth-maps [27], and meshes [16], the popularity of implicit representation [28,48,29] has recently increased. These representations [48,29] serve as a continuous representation on the volumetric occupancy grid [28] to encode

the iso-surface of a shape. Therefore, given any query 3D point, it returns its binary occupancy in the encoded iso-surface. This continuous representation is extremely useful when the input shapes are partial and the spatial locations can be queried on missing regions based on the available regions.

Learning-based 3D Shape Completion. 3D reconstruction and completion of shapes, especially for human 3D body scans, is tackled in different ways [14,43,40,39,15,46,47,13]. Some methods deal with partial point clouds as input [15,46,47,13]. A subset of these methods do not consider textured scans and focus on different types of shape quantization – *e.g.*, voxel grids [15], octrees, and sparse vectors [46] with the aim of capturing only fine geometric details. Furthermore, many of these models [46,7,13] cannot always predict a single shape corresponding to the partial input shape. Another traditional way for body shape completion from a set of partial 3D scans is by non-rigidly fitting a template mesh [4,11,6]. However, improper scaling [4], computational speed [6], and articulated body pose matching [11] is a major problem for these methods.

More closely related works to ours are [9,36,8]. Saint *et al.* [36] and Chibane *et al.* [9] solve the same 3D textured shape completion, where the former uses 3D-Coded [12] and non-rigid refinement to reconstruct the shape and the latter uses deep implicit functions [8]. While [36] provides a fixed UV-parametrization useful for the texture completion, it cannot recover extreme partial body scans due to its restriction to a template body model [25]. On the other hand, [8] can recover coarse-to-fine geometrical details using multiscale implicit representations. However, the implicit representations cannot preserve the UV-parametrization of the input partial shape, which restricts the texture completion to a vertex-based solution [9]. TSCom-Net builds on top the implicit shape representation of [8,9] and extends it to produce higher-resolution texture.

Learning-based 3D Textured Shape Completion. Completing the texture of 3D body shapes is a challenging task due to many factors – *e.g.*, varying shades, unknown clothing boundaries, and complex styles and stripes. This task can be either tackled by completing the RGB texture for each vertex of the completed mesh [9], or directly completing the texture-atlas [36,10]. While the former takes advantage of the shape structure of the vertices to predict textures, its resolution depends on the resolution of the shape, which is often limited due to memory constraints. On the other hand, the texture-atlas completion allows for high-resolution texture generation, but does not take advantage of the structure of the shape when no fixed UV-parametrization is given [20,36], or multiple charts are provided [36,10]. To address these limitations, TSCom-Net uses both paradigms. Firstly, it completes the vertex-based textures, then refines them in the texture-atlas image space using a dedicated inpainting module.

3 Proposed Approach – TSCom-Net

Given a 3D partial body scan $\mathbf{X} = (\mathcal{V}, \mathcal{T}, \mathcal{E})$, tuple defining set of vertices \mathcal{V} , triangles \mathcal{T} , and edges \mathcal{E} along with its corresponding partial texture \mathcal{A} , our aim is to recover the 3D complete body scan \mathbf{Y} with high-resolution texture.

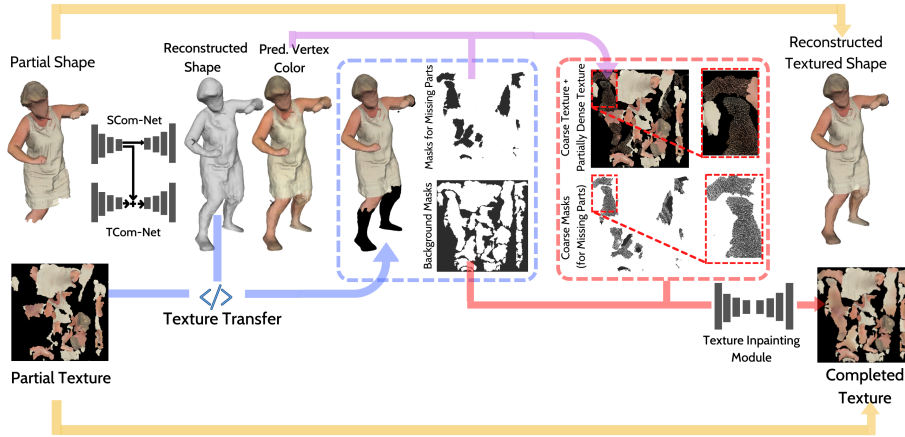


Fig. 2. Overview of TSCom-Net. TSCom-Net has two main components – (I) an end-to-end trainable vertex-based shape and color completion network (*on the left*) that returns low resolution and often over-smoothed vertex colors. The underlying architecture of this modules is based on IFNet [8] and IFNet-Texture [9] along with newly added modifications to boost both the shape and color completion jointly. (II) A coarse-to-fine texture inpainting module (using 2D partial convolutions [23]) that takes partially textured and fully reconstructed body shape, background coarse masks, and coarse-texture and its masks for the missing regions as input, and outputs a high-resolution, complete ‘texture atlas’.

To achieve that, we propose a deep learning-based framework TSCom-Net with two intermediate stages (see Fig. 2). First, we reconstruct the shape and coarse vertex texture of the given partial scan. Then, we further inpaint the partial texture atlas with another network in a coarse-to-fine manner.

3.1 Joint Shape and Texture Completion

Let $\mathcal{X}_s = \mathbb{R}^{N \times N \times N}$ be the discretized voxel grid of the input partial scan \mathbf{X} with resolution N . Each occupied voxel has the value of 1 (0 otherwise). Both the ground-truth and partial scans are normalized. Similarly, another voxel grid, \mathcal{X}_c , is constructed with the texture of the input partial scan \mathbf{X} . Each occupied voxel of \mathcal{X}_c has an RGB texture value in $[0, 1]^3$ obtained from the partial texture ($\langle -1, -1, -1 \rangle$ otherwise).

TSCom-Net consists of two implicit feature networks SCom-Net and TCom-Net that learn to complete the shape and texture of a partial surface, respectively. Different from [9], the two sub-networks are trained jointly to enable the interaction between shape and texture learning. An early fusion technique is utilized to improve the color accuracy by incorporating the shape features in addition to the color features of the partial scan. Consequently, the texture network TCom-Net learns to use the shape information extracted by SCom-Net to predict a more

accurate texture. In what follows, we start by describing the shape completion, then present the texture completion.

Shape Completion: The input discretized voxel grid \mathcal{X}_s is encoded as in [8] using a 3D convolutional encoder g_s to obtain a set of n multiscale shape features,

$$g_s(\mathcal{X}_s) := \mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_n, \quad (1)$$

where $\mathbf{S}_i \in \mathbb{R}^{d_i \times K \times K \times K}$, $1 \leq i \leq n$, denote the shape features with resolution $K = \frac{N}{2^{i-1}}$ and channel dimension $d_i = d_1 \times 2^{i-1}$. Accordingly, \mathbf{S}_n would have the lowest resolution but the highest channel dimensionality among other shape features. Given the features \mathbf{S}_i , the completion of the shape is achieved by predicting the occupancies of the query points $\{\mathbf{p}_j\}_{j=1}^M$, where $\mathbf{p}_j \in \mathbb{R}^3$. At training time, the points \mathbf{p}_j are sampled from the ground-truth \mathbf{Y} . During inference time, \mathbf{p}_j are the centroids of all voxels in \mathcal{X}_s . Multiscale features $\mathbf{S}_{i,j}^p := \phi(\mathbf{S}_i, \mathbf{p}_j)$ can be extracted for each point \mathbf{p}_j using a grid sampling function ϕ , taking and flattening the features of \mathbf{S}_i around the neighborhood of \mathbf{p}_j .

Finally, \mathbf{p}_j and $\mathbf{S}_{i,j}^p$ are decoded with f_s consisting of sequential 1D convolutional layers to predict the occupancy value s_j of \mathbf{p}_j ,

$$f_s(\mathbf{p}_j, \mathbf{S}_{1,j}^p, \mathbf{S}_{2,j}^p, \dots, \mathbf{S}_{n,j}^p) := s_j. \quad (2)$$

The completed mesh structure $\hat{\mathbf{X}}$ is obtained by applying marching cubes [26] on the voxel grid \mathcal{X}_s with the predicted occupancy values s_j .

Texture Completion: For the texture completion, the colored voxel grid \mathcal{X}_c is encoded using a 3D convolutional encoder g_c to obtain a set of m multiscale texture features,

$$g_c(\mathcal{X}_c) := \mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_m, \quad (3)$$

where $\mathbf{C}_i \in \mathbb{R}^{r_i \times L \times L \times L}$, $1 \leq i \leq m$, denote the texture features with resolution $L = \frac{N}{2^{i-1}}$ and channel dimension $r_i = r_1 \times 2^{i-1}$. Texture completion is carried out by predicting the RGB values of the query points $\{\mathbf{q}_j\}_{j=1}^M$, where $\mathbf{q}_j \in \mathbb{R}^3$. The points \mathbf{q}_j are sampled from the ground-truth \mathbf{Y} during training. At the inference time, \mathbf{q}_j are the vertices of the reconstructed mesh $\hat{\mathbf{X}}$. Similar to the shape completion, the grid sampling function $\phi(\mathbf{C}_i, \mathbf{q}_j)$ allows extracting multiscale texture features $\mathbf{C}_{i,j}^q$ for each point \mathbf{q}_j .

The shape and texture completions are fused at the texture decoder level. This is achieved by concatenating the multiscale shape and texture features before feeding them to a decoder f_c consisting of sequential 1D convolutional layers. In particular, the RGB values \mathbf{c}_j for each point \mathbf{q}_j are given by,

$$f_c(\mathbf{q}_j, \mathbf{S}_{1,j}^q, \mathbf{S}_{2,j}^q, \dots, \mathbf{S}_{n,j}^q, \mathbf{C}_{1,j}^q, \mathbf{C}_{2,j}^q, \dots, \mathbf{C}_{m,j}^q) := \mathbf{c}_j. \quad (4)$$

Finally, vertex textures are obtained by attaching the predicted \mathbf{c}_j to $\hat{\mathbf{X}}$.

3.2 Texture Refinement

The proposed joint shape and texture networks are able to predict the vertex textures of a completed mesh. However, the predicted vertex textures have two

limitations: (1) they are predicted for the full body, including the existing regions of the partial body. Consequently, high-level texture details of input partial body might be lost in the predicted textured body mesh; (2) the resolution of the vertex textures depends on the resolution of the reconstructed shape. This implies that high-resolution vertex textures come at a cost of high-resolution predicted shape, which is not straightforward to obtain due to memory constraints.

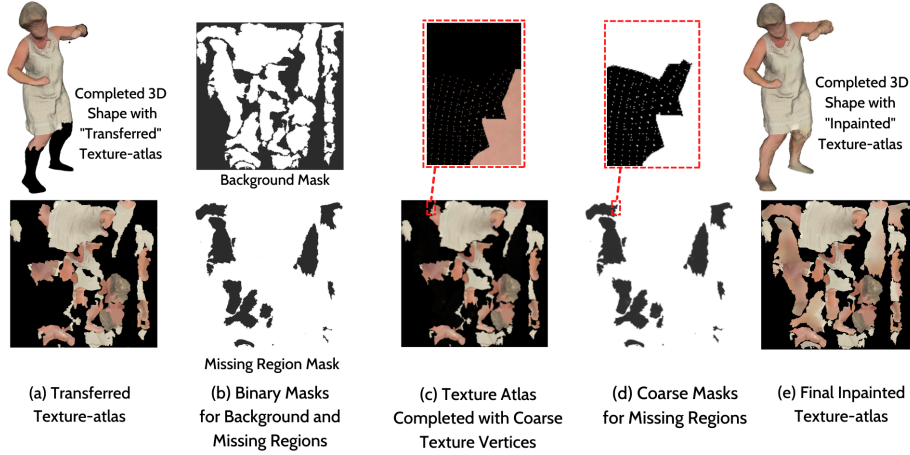


Fig. 3. Texture Refinement. (a) the texture-atlas is transferred to the completed 3D shape, (b) the masks for the missing regions and background are identified, (c) the vertex textures are projected into the transferred texture-atlas, (d) the masks for missing regions are updated by unmasking the regions of the projected vertex textures, (e) the final inpainted texture-atlas.

To overcome these issues, we use a *texture-atlas* [21,36] based refinement of the predicted vertex textures. This refinement reuses the coarse vertex textures predicted by the joint implicit shape and texture network and refines them in the 2D image space, while preserving the original texture from the partial input scan. In particular, the original texture of the partial scan is transferred to the completed shape by a ray-casting algorithm, as in [36]. This allows the creation of a UV map and a texture atlas \mathcal{A} for the completed mesh as depicted in Fig. 3(a). Following [36], the missing regions and the background regions are identified in the transferred texture atlas to create the two binary masks M and M_b shown in Fig. 3(b). Using the UV map created by the texture transfer, the vertex textures are projected to obtain a coarsely completed texture-atlas \mathcal{A}_c as sketched in Fig. 3(c). The mask for missing regions M is then updated with the projected vertex textures, yielding a coarse mask M_c as displayed in Fig. 3(d).

Given the coarsely completed texture-atlas \mathcal{A}_c , the coarse mask of missing regions M_c , and the background mask M_b , the problem of texture refinement is formulated as an image inpainting one. Specifically, we opt for the texture-atlas

inpainting method proposed in [36], which adapts partial convolution inpainting [23] to the context of texture-atlas. Partial convolutions in [23,36] extend standard convolutions to convolve the information from unmasked regions (*i.e.* white regions in the binary masks). Formally, let us consider a convolution filter defined by the weights w , the bias b and the feature values \mathcal{A}_c^w of the texture-atlas \mathcal{A}_c for the current sliding window. Given the coarse mask of missing regions M_c and the corresponding background mask M_b , the partial convolution at every location, similarly defined in [36], is expressed as,

$$a_c = \begin{cases} w^T(\mathcal{A}_c^w \odot M_c \odot M_b) \cdot \frac{\text{sum}(\mathbf{1})}{\text{sum}(M_c \odot M_b)} + b & \text{if } \text{sum}(M_c \odot M_b) > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where \odot denotes element-wise multiplication, and $\mathbf{1}$ has same shape as M_c but with all elements being 1. As proposed in [36], the masks M_c are updated after every partial convolution, while the background masks are passed to all partial convolutions layers without being updated by applying *do-nothing* convolution kernels. The partial convolutional layers are employed in a UNet architecture [33] instead of standard convolutions. At training time, the vertex textures are sampled from the ground-truth texture-atlas. The same loss functions in [23,36] are used to train the network. It is important to highlight that the proposed texture refinement is different from [36] as it reuses the predicted vertex textures instead of inpainting the texture-atlas from scratch. We show in Section 4 that the proposed refinement outperforms the inpainting from scratch and the vertex texture based completion. Furthermore, we reveal that such refinement can be used to improve other vertex based texture completion.

4 Experimental Results

Dataset and Evaluation Metrics. Our method was trained on the 3DBody-*Tex.v2* dataset [34,35,37] which has been recently used as a benchmark for the SHARP challenge [3]. The dataset contains a large variety of human poses and different clothing types from 500 different subjects. Each subject is captured with 3 different poses and different clothing types, such as close-fitting or arbitrary casual clothing. The number of ground truth scans in the training set is 2094. A number of 15904 partial scans for training and validation were generated using the routines provided by the SHARP challenge organizers [2]. The number of unseen (during training) scans in the evaluation set is 451.

As considered in SHARP challenges, the evaluation is conducted in terms of shape scores S_s , texture scores S_t , and the area scores S_a . Shape and texture scores are calculated via measuring surface-to-surface distances by sampling points from the ground truth and the reconstructed meshes. The area score estimates the similarity between the triangle areas of the meshes. The final score S_r is calculated as $S_r = \frac{1}{2}S_a(S_s + S_t)$. Details about these metrics can be found

in [37,2]². The evaluation of the completed meshes is performed via the Codalab system provided by the SHARP challenge³.

4.1 Network Training Details

We trained the joint implicit networks using the Adam optimizer with a learning rate of 10^{-4} for 54 epochs. The model was trained on an NVIDIA RTX A6000 GPU using the Pytorch library. The query points for the training are obtained from the ground truth surfaces by sampling 100000 points. During training, we sub-sample 50000 of these points at each iteration. Gaussian random noise $\mathcal{N}(\sigma)$ is added to each point to move the sampled point near or far from the surface, depending on the σ value. Similar to [9], the σ is chosen as 0.01 for half of the points and 0.1 for the other half. We did not add noise to the query points of the texture network, as its goal is to predict the color value of the points sampled from the surface. Partial scans are voxelized by sampling 100000 points from the partial surface and setting the occupancy value in the nearest voxel grid to 1. Similarly for the colored voxelization, the value of the nearest voxel is set to the RGB value of the sampled point obtained from the corresponding texture-atlas. The input voxel resolution is 128 and the resolution for the final retrieval is 256.

We follow a similar naming convention as in [45] for our architecture details. Let $c3-k$ denote Conv3D-ReLU-BatchNorm block and $d3-k$ denote two Conv3D-ReLU blocks followed by one BatchNorm layer where k is the number of filters. The kernel size for all 3D convolutional layers is 3×3 . Let $gs-i$ represent the grid feature sampling of the query points from the output of the previous layer, and mp denote 3D max pooling layer. Let $c1-k$ denote the Conv1D layer with 1×1 kernel where k is the number of output features and ReLU activation except the output layer where the activation is linear. The encoder architecture for shape and texture networks is composed of the following layers: $gs-0$, $c3-16$, $gs-1$, mp , $d3-32$, $d3-32$, $gs-2$, mp , $d3-64$, $d3-64$, $gs-3$, mp , $d3-128$, $gs-4$, mp , $d3-128$, $d3-128$, $gs-5$.⁴ The decoder architecture for the shape network contains: $c1-512$, $c1-256$, $c1-256$, $c1-1$. The decoder architecture for the texture network consists of $c1-512$, $c1-256$, $c1-256$, $c1-3$. The partial convolutional network is trained with Adam optimizer and learning rate of 10^{-4} for 330000 iterations. The original texture size of 2048×2048 was used for training the network with a batch size of 1.

4.2 Results and Evaluation

In this section, a qualitative and quantitative analysis of the results of TSCom-Net against the results of the SHARP challenge participants are presented. Other

² The details about the metrics can be accessed via: <https://gitlab.uni.lu/cvi2/cvpr2022-sharp-workshop/-/blob/master/doc/evaluation.md>

³ The leaderboard on Codalab can be accessed via: <https://codalab.lisn.upsaclay.fr/competitions/4604#results>

⁴ Sampling features $gs-1$, $gs-2$, ..., $gs-5$ are flattened and concatenated. Shape features are also added here for the texture encoding.

Method	Shape Score(%)	Area Score(%)	Texture Score(%)	Final Score(%)
IFNet-Texture[9]	85.44 ± 2.93	96.26 ± 6.35	81.25 ± 7.61	83.34 ± 6.86
Method Raywit	85.91 ± 7.14	93.96 ± 3.96	83.45 ± 8.43	84.68 ± 7.63
Method Rayee	86.13 ± 7.32	96.26 ± 3.61	83.23 ± 8.31	84.68 ± 7.74
Method Janaldo	89.76 ± 4.97	96.76 ± 2.28	87.10 ± 6.33	88.43 ± 5.56
TSCom-Net (Ours)	85.75 ± 6.15	96.68 ± 2.89	83.72 ± 6.95	84.73 ± 6.5

Table 1. Quantitative Results for SHARP 2022. The best and second best scores are denoted in **bold-black** and **bold-gray** colors respectively.

participants of the challenge employ implicit networks for the shape reconstruction with vertex-based texture completion [3]. We also compare our method to the state-of-the-art IFNet-Texture [9] method, which won the previous editions of SHARP 2021 and 2020 [37]. The shape network of [9] is re-trained with the generated partial data of SHARP 2022. For the vertex texture predictions, the available pretrained model provided by the authors was used.

Quantitative Evaluation: In Table 1, we illustrate the quantitative results using the metrics of SHARP challenge [37]. Overall, our method is ranked second in the leaderboard with a final score of 84.73%, obtaining a better texture score than Method-Rayee, Method-Raywit and IFNet-Texture [9]. It should be noted that the texture score depends on the shape score as well, since it is not possible to get a correct texture score for an incorrectly predicted shape.

Method	Texture Score (%)
Method-Janaldo	87.10 ± 6.33
Method-Janaldo + Our Texture Refinement	87.54 ± 6.19

Table 2. Effectiveness of Our Texture Refinement.

Furthermore, we demonstrate the effectiveness of our texture refinement method by applying it to the predictions of Method-Janaldo. The results in Table 2 show that our inpainting network introduced in Section 3.2 improves the texture score by 0.44%. This shows that the proposed texture refinement can be used as an additional component to improve other shape and texture completion methods that predict low-resolution vertex textures.

Finally, Fig. 4 show the distributions of shape, texture, and final scores for the predictions of TSCom-Net, Method-Janaldo, Method-Janaldo with our texture refinement, and IFNet-Texture [9]. The first row of this figure illustrates the overall distribution of the scores and highlights the superiority of our texture scores *w.r.t.* IFNet-Texture [9]. While Method-Janaldo outperforms our results, it can be observed that if we endow it with our texture refinement scheme, the distribution of texture scores becomes slightly higher. In the second and third

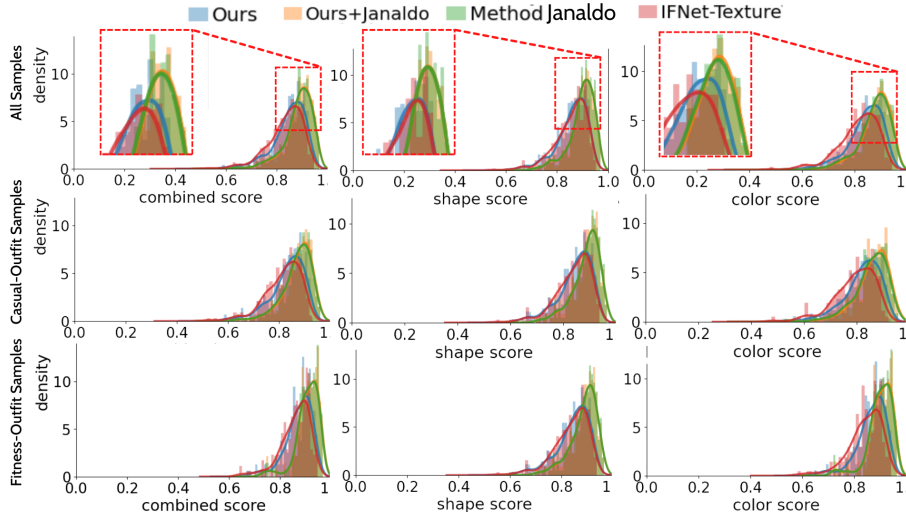


Fig. 4. Score Distribution. Distribution of shape, color and combined final scores ($\frac{\text{score}}{100}$ along x-axes of the plots). The first row reports the distribution of the scores on all the scans. The second row focuses on casual-outfit scans while the third one reports the distributions on fitness-outfit scans.

row of Fig. 4, we report the distribution of the scores on scans with casual-outfit and fitness-outfit. Unsurprisingly, casual-outfit scans were more challenging than fitness-outfit ones for all methods. Nevertheless, our approaches recorded more improvements *w.r.t.* Method-Janaldo and IFNet-Texture on the casual-outfit scans than the fitness-outfit ones.

It is important to mention that the reported scores are the result of mapping distance values (shape or texture) to scores in percentage using a parametric function. This mapping might make the scores from different methods close to each other, depending on the chosen parameters. We note that the original parameters of the SHARP challenge 2022 metrics are used. Thus, we further conducted a qualitative evaluation of our method.

Qualitative Evaluation: Fig. 5 shows a qualitative comparison of our approach to other competing methods on models with casual outfit and fitness outfit. Considering the coat missing region of the top row model, it can be noted that IFNet-Texture [9] and Method-Raywit are unable to predict the correct colors. On the other hand, Method-Rayee and Method-Janaldo produce over-smoothed textures, creating blurry artifacts. Neither of these effects can be observed on the TSCom-Net predictions. In the second row, the bottom legs and feet appear to be the most difficult regions to recover. Method-Rayee and Method-Raywit fail to produce the correct shape for these missing regions. While IFNet-Texture [9] and Method-Janaldo are able to recover the correct shape, both generate white color artifacts on the jeans. Our method is the only one to produce more reasonable texture predictions, showing a sharper change in color between the jeans and

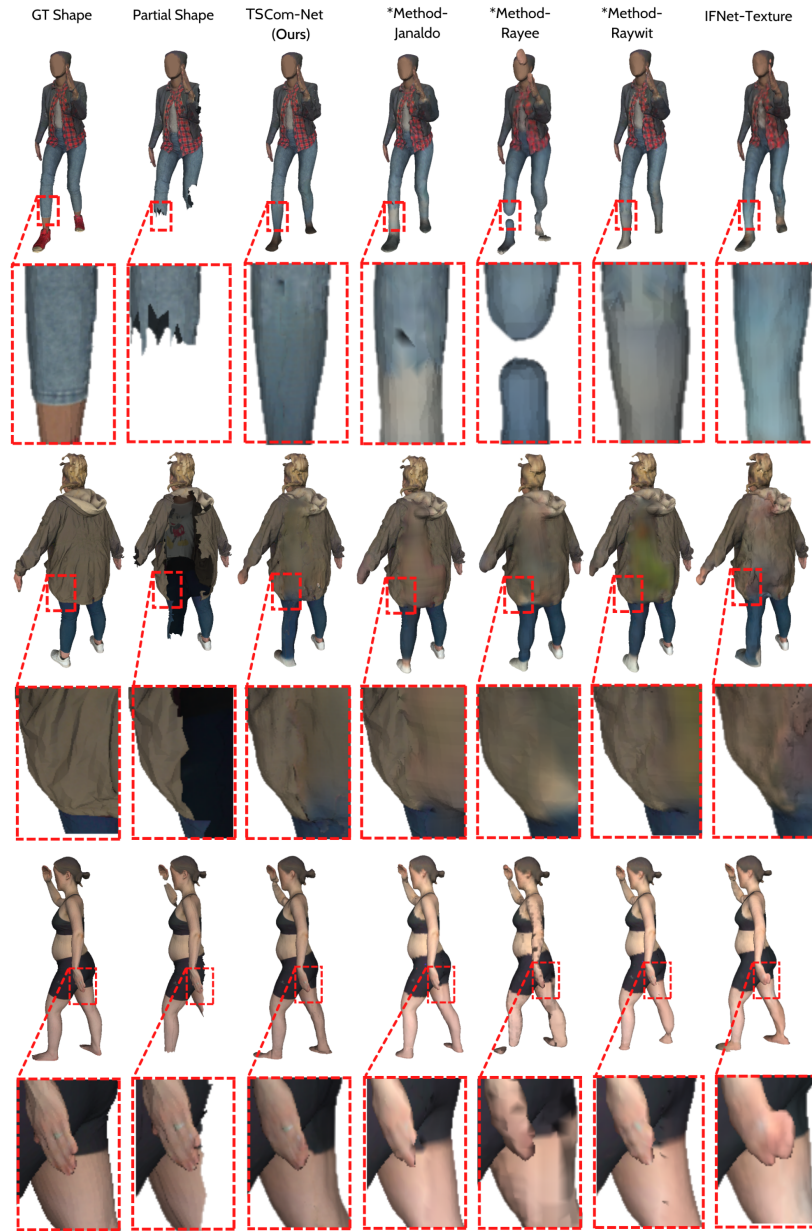


Fig. 5. Qualitative results. Visual comparisons of textured body shape completion results by different competing methods (as per Table.1). The *first two rows* depict results on samples with casual outfits, *i.e.*, more variation on garment texture pattern. The *last two rows* depict results on samples with fitness outfits.

the feet. Similar to the second model, it is apparent that our results on the third model are sharper for the regions with a color change from skin to black when compared to the other results. The visual comparisons illustrate that our results are of higher resolution with better texture representation than the competing approaches despite the close quantitative scores.

4.3 Ablation Study

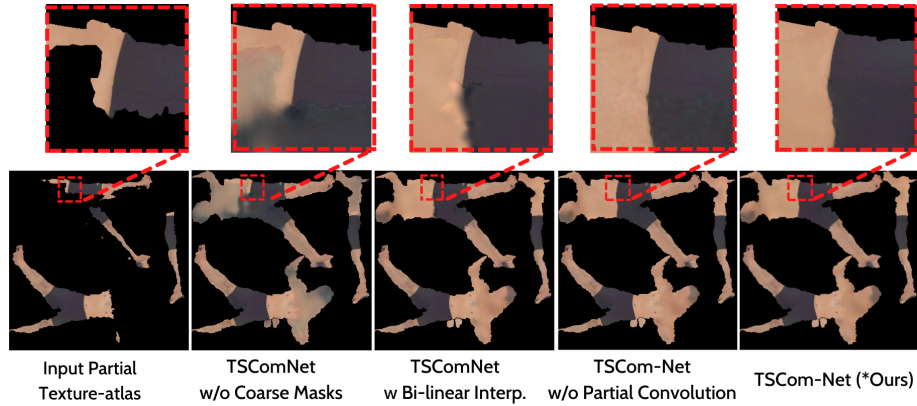


Fig. 6. Multi-stage improvements of texture-atlas inpainting in TSCom-Net.

Method	Shape Score(%)	Area Score(%)	Texture Score(%)	Final Score(%)
IFNet-Texture [9]	85.44 ± 2.93	96.26 ± 6.35	81.25 ± 7.61	83.34 ± 6.86
Texture-transfer Baseline	85.75 ± 6.15	96.68 ± 2.89	56.51 ± 18.98	71.13 ± 11.11
Ours w/o Coarse Masks	85.75 ± 6.15	96.68 ± 2.89	81.04 ± 7.92	83.39 ± 6.87
Ours w/o Tex. Refine.	85.75 ± 6.15	96.68 ± 2.89	83.27 ± 7.08	84.53 ± 6.54
Ours w/ Bilinear Interp.	85.75 ± 6.15	96.68 ± 2.89	83.66 ± 6.95	84.71 ± 6.50
Ours w/o Partial Conv.	85.75 ± 6.15	96.68 ± 2.89	83.68 ± 6.96	84.71 ± 6.50
TSCom-Net (Ours)	85.75 ± 6.15	96.68 ± 2.89	83.72 ± 6.95	84.73 ± 6.50

Table 3. Quantitative scores of TSCom-Net with multi-stage improvements.

Our joint implicit function learning for shape and texture (TSCom-Net w/o Texture Refinement) gives a 1.19% (cf. Table 3) increase in the final score with respect to [9] demonstrating the effect of early fusion between SCom-Net and TCom-Net. Only transferring the partial texture to the reconstructed shape, where the color values for the missing regions are all black, gives us a baseline texture score of 56.51%. Training an inpainting network directly on the partial textures (TSCom-Net w/o Coarse Masks) increase texture score to 81.04% which is 0.21% lower than [9] and 24.53% higher than the baseline. Conducting bilinear interpolation of the vertex colors in missing regions is giving a texture score of 83.66%. All-in-all partial convolutions, instead of standard convolutions, gives

stable and more sharper texture inpainting results. Fig. 6 depicts how the different types artifacts or lack of sharpness appear when other options of inpainting are tested. Finally, TSCom-Net consisting of all the components is giving the highest final score of 84.73%.

Method	Partiality Type		Shape Score (%)	Texture Score (%)	Final Score (%)
	Training	Testing			
IFNet-Texture [9]	T2	T1	87.99 ± 4.65	84.32 ± 5.73	86.15 ± 5.09
IFNet-Texture [9]	T1	T1	86.49 ± 3.96	89.28 ± 2.89	87.88 ± 3.32
3DBooster [36]	T1	T1	58.81 ± 14.99	72.31 ± 6.79	65.57 ± 3.32
TSCom-Net (Ours)	T2	T1	88.05 ± 4.84	85.46 ± 5.56	86.75 ± 5.14

Table 4. Generalizability of TSCom-Net when trained and tested on samples with different types of partiality.

Generalization to Other Types of Partiality: In this section, we evaluate the generalization capability of our model to other types of partiality. In particular, we consider the *view-based* partiality (T2) introduced in SHARP 2022 [3] to train the models and use the *hole-based* partiality (T1) introduced in previous editions of SHARP [37] for inference. This implies that the networks trained on (T2) have never seen hole-based partial scans (T1). Table 4 demonstrates a comparison of the generalization capability of our method to IFNet-Texture [9] that is also trained on (T2) and tested on (T1) subset. Following these settings, we obtain an increase of 1.14% for the texture score compared to [9]. We also compare our model trained on (T2) and tested on (T1) to IFNet-Texture [9] and 3DBooster [36] both trained on (T1) and tested on (T1). In this case, our approach significantly outperforms [36] while achieving comparable results to [9] although both were trained on (T1).

5 Conclusion

This paper presents a method for completing the shape and texture of 3D partial scans. Joint implicit feature networks are proposed for learning to complete the shape and textures. Moreover, a new coarse-to-fine texture refinement network was introduced. It generates high-resolution texture from the predicted coarse vertex texture and the available partial texture. Experimental evaluations show that our method gives visually more appealing results than the state-of-the-art and is positioned second in the SHARP 2022 challenge. In future, we plan to make the entire TSCom-Net modules end-to-end trainable for completing 3D scans and refining the texture. At the same time, we will investigate neural implicit radiance field for texture completion (with editable 2D UV texture map) and 3D surface reconstruction.

References

1. SHARP 2021, the 2nd shape recovery from partial textured 3d scans. <https://cvi2.uni.lu/sharp2021/>, accessed: 2022-07-23
2. SHARP 2022 Repository, the repository of the 3rd shape recovery from partial textured 3d scans. <https://gitlab.uni.lu/cvi2/cvpr2022-sharp-workshop>, accessed: 2022-07-23
3. SHARP 2022, the 3rd shape recovery from partial textured 3d scans. <https://cvi2.uni.lu/sharp2022/>, accessed: 2022-07-23
4. Ali, S.A., Golyanik, V., Stricker, D.: Nrga: Gravitational approach for non-rigid point set registration. In: International Conference on 3D Vision (3DV) (2018)
5. Ali, S.A., Kahraman, K., Reis, G., Stricker, D.: Rpsrnet: End-to-end trainable rigid point set registration network using barnes-hut 2d-tree representation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
6. Ali, S.A., Yan, S., Dornisch, W., Stricker, D.: Foldmatch: Accurate and high fidelity garment fitting onto 3d scans. In: IEEE International Conference on Image Processing (ICIP) (2020)
7. Arora, H., Mishra, S., Peng, S., Li, K., Mahdavi-Amiri, A.: Multimodal shape completion via implicit maximum likelihood estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
8. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
9. Chibane, J., Pons-Moll, G.: Implicit feature networks for texture completion from partial 3d data. In: European Conference on Computer Vision (ECCV) Workshops (2020)
10. Deng, J., Cheng, S., Xue, N., Zhou, Y., Zafeiriou, S.: Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7093–7102 (2018)
11. Golyanik, V., Reis, G., Taetz, B., Stricker, D.: A framework for an accurate point cloud based registration of full 3d human body scans. In: International Conference on Machine Vision and Applications (ICMVA) (2017)
12. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3d-coded: 3d correspondences by deep deformation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
13. Gurusurthy, S., Agrawal, S.: High fidelity semantic shape completion for point clouds using latent optimization. In: IEEE Winter Conference on Applications of Computer Vision (WACV) (2019)
14. Han, X.F., Laga, H., Bennamoun, M.: Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**(5), 1578–1604 (2019)
15. Han, X., Li, Z., Huang, H., Kalogerakis, E., Yu, Y.: High-resolution shape completion using deep neural networks for global structure and local geometry inference. In: IEEE International Conference on Computer Vision (ICCV) (2017)
16. Hanocka, R., Hertz, A., Fish, N., Giryas, R., Fleishman, S., Cohen-Or, D.: Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)* **38**(4), 1–12 (2019)
17. Hasler, N., Ackermann, H., Rosenhahn, B., Thormählen, T., Seidel, H.P.: Multi-linear pose and body shape estimation of dressed subjects from image sets. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010)

18. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end recovery of human shape and pose. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
19. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: European Conference on Computer Vision (ECCV) (2018)
20. Lazova, V., Insafutdinov, E., Pons-Moll, G.: 360-degree textures of people in clothing from a single image. In: International Conference on 3D Vision (3DV) (2019)
21. Lévy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics (TOG)* **21**(3), 362–371 (2002)
22. Li, X., Liu, S., Kim, K., Mello, S.D., Jampani, V., Yang, M.H., Kautz, J.: Self-supervised single-view 3d reconstruction via semantic consistency. In: European Conference on Computer Vision (ECCV) (2020)
23. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: European Conference on Computer Vision (ECCV) (2018)
24. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems* **32** (2019)
25. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)* **34**(6), 1–16 (2015)
26. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics* **21**(4), 163–169 (1987)
27. Malik, J., Abdelaziz, I., Elhayek, A., Shimada, S., Ali, S.A., Golyanik, V., Theobalt, C., Stricker, D.: Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
28. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
29. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
30. Patel, P., Huang, C.H.P., Tesch, J., Hoffmann, D.T., Tripathi, S., Black, M.J.: Agora: Avatars in geography optimized for regression analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13468–13478 (2021)
31. Prokudin, S., Black, M.J., Romero, J.: Smplpix: Neural avatars from 3d human models. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1810–1819 (2021)
32. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
33. 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 (2015)
34. Saint, A., Ahmed, E., Cherenkova, K., Gusev, G., Aouada, D., Ottersten, B., et al.: 3dbodytex: Textured 3d body dataset. In: International Conference on 3D Vision (3DV) (2018)

35. Saint, A., Cherenkova, K., Gusev, G., Aouada, D., Ottersten, B., et al.: Bodyfitr: robust automatic 3d human body fitting. In: IEEE International Conference on Image Processing (ICIP) (2019)
36. Saint, A., Kacem, A., Cherenkova, K., Aouada, D.: 3dbooster: 3d body shape and texture recovery. In: European Conference on Computer Vision (ECCV) (2020)
37. Saint, A., Kacem, A., Cherenkova, K., Papadopoulos, K., Chibane, J., Pons-Moll, G., Gusev, G., Fofi, D., Aouada, D., Ottersten, B.: Sharp 2020: The 1st shape recovery from partial textured 3d scans challenge results. In: European Conference on Computer Vision (ECCV) (2020)
38. Sarkar, K., Varanasi, K., Stricker, D.: Learning quadrangulated patches for 3d shape parameterization and completion. In: International Conference on 3D Vision (3DV) (2017)
39. Sinha, A., Unmesh, A., Huang, Q., Ramani, K.: Surfnet: Generating 3d shape surfaces using deep residual networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
40. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: IEEE International Conference on Computer Vision (ICCV) (2017)
41. Tian, Y., Zhang, H., Liu, Y., Wang, L.: Recovering 3d human mesh from monocular images: A survey. arXiv preprint arXiv:2203.01923 (2022)
42. Wang, J., Zhong, Y., Li, Y., Zhang, C., Wei, Y.: Re-identification supervised texture generation. In: IEEE Computer Vision and Pattern Recognition (CVPR) (2019)
43. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: European Conference on Computer Vision (ECCV) (2018)
44. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)* **36**(4), 1–11 (2017)
45. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
46. Yan, X., Lin, L., Mitra, N.J., Lischinski, D., Cohen-Or, D., Huang, H.: Shapeformer: Transformer-based shape completion via sparse representation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
47. Yuan, W., Khot, T., Held, D., Mertz, C., Hebert, M.: Pcn: Point completion network. In: International Conference on 3D Vision (3DV) (2018)
48. Zheng, Z., Yu, T., Dai, Q., Liu, Y.: Deep implicit templates for 3d shape representation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
49. Zheng, Z., Yu, T., Wei, Y., Dai, Q., Liu, Y.: Deephuman: 3d human reconstruction from a single image. In: IEEE International Conference on Computer Vision (ICCV) (2019)