3DBooSTeR:
3D Body Shape and Texture Recovery

Alexandre Saint[1], Anis Kacem[1], Kseniya Cherenkova[1,2], and Djamila Aouada[1]

[1] SnT, University of Luxembourg
{alexandre.saint,anis.kacem,djamila.aouada}@uni.lu
[2] Artec3D
kcherenkova@artec-group.com

Abstract. We propose 3DBooSTeR, a novel method to recover a textured 3D body mesh from a textured partial 3D scan. With the advent of virtual and augmented reality, there is a demand for creating realistic and high-fidelity digital 3D human representations. However, 3D scanning systems can only capture the 3D human body shape up to some level of defects due to its complexity, including occlusion between body parts, varying levels of details, shape deformations and the articulated skeleton. Textured 3D mesh completion is thus important to enhance 3D acquisitions. The proposed approach decouples the shape and texture completion into two sequential tasks. The shape is recovered by an encoder-decoder network deforming a template body mesh. The texture is subsequently obtained by projecting the partial texture onto the template mesh before inpainting the corresponding texture map with a novel approach. The approach is validated on the 3DBodyTex.v2 dataset.

Keywords: 3D shape completion · Human body shape · Point cloud · Texture · Inpainting

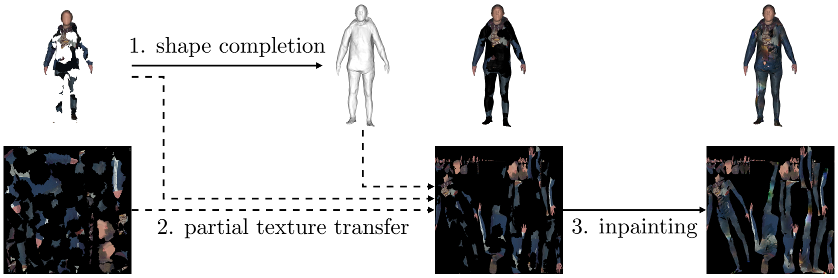


Fig. 1: Overview of the proposed approach for completing a partial textured 3D body mesh. 1) The complete shape is estimated. 2) The partial texture is transferred onto the estimated shape. 3) The corresponding texture image is inpainted.

1 Introduction

The completion of a partial textured 3D body shape is key to enable the digital representation of 3D body shape with realistic details, in a reasonable time and in an automated way. This is required in applications such as virtual reality. Capturing the 3D human shape and texture of the human body is complex due to the occlusion of body parts, the complex shape of the clothing wrinkles, the variation of the pose in time, etc. Some photogrammetric scanner systems, on the one hand, use a large array of cameras to cover the body shape from all possible angles. These systems are able to capture textured 3D shape sequences at high frame rates (120kHz). However, they still suffer from occlusion and cannot represent fine details, such as the fingers and the ears. Hand-held scanners, on the other hand, can be brought to resolve fine details and cover all view angles, but they require a static target. For an unconstrained usage, it is then more desirable to acquire a partial shape and come with an effective completion solution that recovers the missing data. Aware of this emergent need, some state-of-art works have tried to recover the missing information provided by the scanning devices. Most of them consider the problems of shape and texture completion independently. Accordingly, the relevant literature is presented separately below.

Shape completion A simple approach to shape completion is the hole-filling algorithm (*e.g.* Davis et al. [4]), in which the missing shape regions are filled with a surface patch joining the boundaries of the available surface. This approach is limited to relatively small holes with respect to the surrounding surface and to relatively smooth regions.

An approach to regularise the shape completion is to rely on a template shape that is deformed to match the input partial shape. Szeliski et al. [26] deform a simple convex shape to represent anatomical body parts such as the head. Anguelov et al. [1] learn a parametric model of the human body pose and shape to regularise the completion of a full 3D body shape while handling the large deformations caused by variation in pose. This sort of methods usually requires manual initialisation, as shown by Saint et al. [24, 22, 23], who propose fully-automatic body model fitting approaches by exploiting the colour or texture information in human scans, as available in 3DBodyTex [22]. This allows recovering shapes with relatively large proportions of missing data. The body models based on SMPL provide a reasonable trade-off between computational efficiency and expressiveness [16, 19]. However, body models smooth out the shape details and are limited to the body shape without clothing. In these works, the completion of the texture is not considered, even though the texture might be used to regularise the shape completion [22, 23].

Some works use volumetric convolutions to complete partial 3D shapes [28, 3, 10]. The achievable resolution is limited due to the high computational complexity of 3D convolutions. Moreover, this category of approaches works well on relatively rigid shapes [28, 3] (*e.g.* objects of the same class) but less well on deformable shapes [14] (*e.g.* shape of the human body or of animals). Chibane et al. [2] represent the 3D body shape with an implicit function. The implicit function is approximated by a deep neural network and learned from a dataset of example 3D body shapes. This method allows completing partial shapes but does not consider the texture. It accepts different input 3D shape representations (*e.g.* point cloud, mesh, voxel grid), however, the output surface must be recovered from the implicit function with a post-processing step, such as the marching cube algorithm [17].

Several works learn the space of body shape deformations using deep learning [9, 13, 14, 18]. These models use encoder-decoder architectures. An input shape is encoded into a latent representation by the encoder. The decoder then deforms a base body mesh from a canonical pose and shape to a specific pose and identity, using the intermediate representation as input. The deformations are performed with mesh convolutions (*e.g.* PointNet [20] or FeaStNet [27]). The parameters of the network are learned from a dataset of example 3D body shapes. Some works target only the body shape with minimal close-fitting clothing [9, 13, 14], which is locally smooth and regular. Other works target the body shape with casual clothing [18], which contains irregular local variations, such as wrinkles, due to factors including the cut and the fabric. The completion of texture information is not tackled in these works.

Texture completion Deng et al. [6] recover the 3D shape and the colour information of a face from a non-frontal 2D view. Shape and texture completion are decoupled. A 3D morphable model (3DMM) is first fitted to the image. Then, the available colour information is projected onto the UV map of the template mesh. Finally, the UV map is inpainted to recover the missing colour information. The fitting of a body is more complex due to the pose variation, larger and more non-linear than the variation in the expression. The UV map for the face is a single chart. For a body model, it is a set of multiple charts representing different body regions.

In the context of partial shape and texture completion, we present our approach to solve the SHARP challenge [25] on recovering large regions of partial textured body meshes. This challenge provides 3DBodyTex.v2, a dataset with thousands of textured body meshes. The training set and the validation set contain the ground-truth textured body meshes. The evaluation set contains only partial meshes. The goal is to estimate the complete shapes in the evaluation set.

Our contribution, sketched in Fig. 1, is a method, named 3DBooSTeR, to recover a 3D body mesh with a corresponding high-resolution texture from a textured partial 3D body scan. The tasks of shape and texture completion are decoupled into a sequential pipeline. The shape completion method is a data-driven mesh deformation deep learning network (based on Groueix et al. [9]) that produces an output mesh of fixed topology. The texture completion is reduced to an inpainting task of the texture image of the reconstructed mesh. An novel inpainting method (based on Liu et al. [15]) is proposed. It is specifically designed to handle the inpainting of a texture image with robustness to irregularly-shaped incomplete regions and irregularly-shaped background regions that must be ignored.

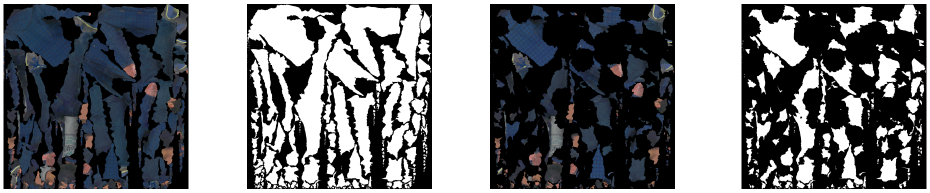
The rest of the paper is organised as follows: Section 2 introduces the problem and the notations used in the paper. Section 3 presents the proposed approach for shape and texture completion. Experimental results are reported in Section 4. In Section 5, we conclude the paper.

2 Problem Statement

A textured body shape is denoted $\mathcal{X} = (\mathcal{S}, T)$, where $\mathcal{S} = (\mathbf{V}, \mathbf{F}, \pi)$ is a body mesh with n_v vertices stacked in a matrix $\mathbf{V} \in \mathbb{R}^{n_v \times 3}$, n_f triangular faces encoded in $\mathbf{F} \in \mathbb{N}^{n_f \times 3}$ as triplets of vertex indices, and a 2D parametrisation, π, defining a mapping of the faces between the 3D shape, $\mathcal{S}$, and the 2D texture image, T.

Given a partial textured body shape $\mathcal{X}_p = (\mathcal{S}_p, T_p)$, with, $\mathcal{S}_p$, the partial mesh, and, T_p, the partial texture, we aim to predict a complete textured body mesh $\hat{\mathcal{X}} = (\hat{\mathcal{S}}, \hat{T})$ that approximates well the ground truth $\mathcal{X} = (\mathcal{S}, T)$. Moreover, the estimation $\hat{\mathcal{X}}$ should preserve the partially provided texture and shape information as much as possible.

Texture atlas A texture image T has a texture atlas structure [12] consisting of a set of charts (*i.e.* small pieces of the body texture) gathered together in a single image. Each of these charts is mapped onto a different region of the 3D mesh $\mathcal{S}$ using the 2D parametrisation π. This allows for densely colouring a 3D mesh from a 2D image. Fig. 2a shows an example texture atlas corresponding to a complete body mesh. Fig. 2b shows its corresponding background mask. The background corresponds to the regions outside of the charts. They do not contain any texture information and are coloured black by convention. The background is defined in a *background mask* M_b where $M_b(i, j) = 0$ if (i, j) is a background pixel, and $M_b(i, j) = 1$ otherwise (foreground). Fig. 2c and 2d, show the partial texture atlas and background mask of a corresponding partial mesh (generated synthetically).



(a) A complete texture atlas.

(b) Background mask of complete texture.

(c) A partial texture atlas.

(d) Background mask of partial texture.

Fig. 2: Example of complete and partial texture atlases with their corresponding background masks M_b. Sample from the 3DBodyTex.v2 dataset.

3 Proposed Approach

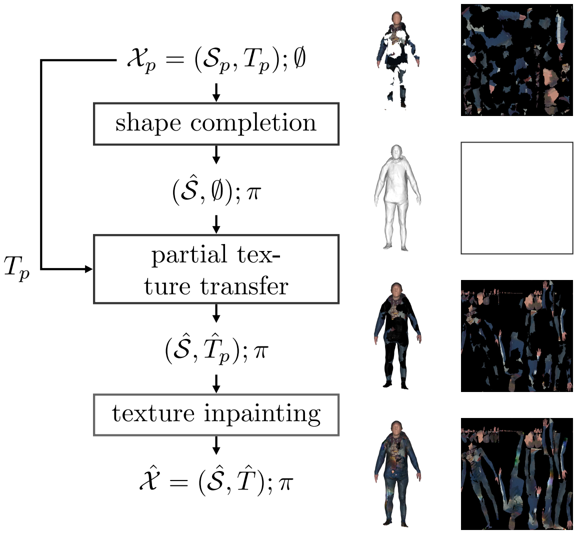

$\mathcal{X}_p = (\mathcal{S}_p, T_p); \emptyset$
shape completion
$(\hat{\mathcal{S}}, \emptyset); \pi$
T_p
partial texture transfer
$(\hat{\mathcal{S}}, \hat{T}_p); \pi$
texture inpainting
$\hat{\mathcal{X}} = (\hat{\mathcal{S}}, \hat{T}); \pi$

Fig. 3: Overview of the proposed approach for 3D body shape and texture completion.

We propose to solve the problem of textured 3D body shape completion with two sequential tasks: shape completion, followed by texture completion (Fig. 3). First, a complete 3D shape $\hat{\mathcal{S}}$ is predicted from the partial shape $\mathcal{S}_p$ by an encoder-decoder model. The encoder-decoder completes the input partial shape by deforming a template mesh of a full 3D body into the corresponding pose and shape of the input. The texture information of the partial input mesh is then projected onto the estimated shape, $\hat{\mathcal{S}}$, to obtain a completed shape with partial texture, $\hat{T}_p$. The regions with missing texture information are then identified on the texture image, $\hat{T}_p$. Given the partial texture and the missing regions, the task of texture completion on a 3D shape is turned into an image inpainting task with additional constraints to handle the specific image representation of texture atlas. Indeed, the texture image contains irregularly-shape background regions that must be correctly ignored to avoid their propagation and unrealistic inpainting results. The different stages of the approach are detailed below.

3.1 3D Body Shape Completion

The 3D body shape completion is performed in two steps, as illustrated in Fig. 4. First, an encoder-decoder network predicts a rough estimation of the pose and

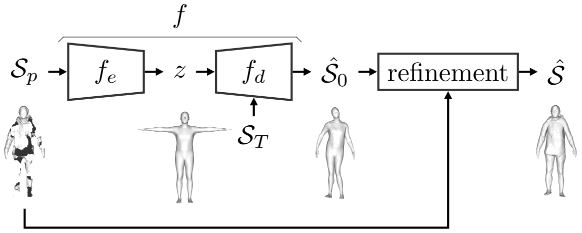


Fig. 4: Pipeline for 3D shape completion: 1) An encoder-decoder network produces a first estimate of a complete shape. 3) The estimate is refined to better fit the clothing.

shape of the partial input shape. Second, the estimation is refined to better match the clothing shape.

The *encoder-decoder*, f, maps a partial 3D shape, $\mathcal{S}_p$, onto a completed shape, $\hat{\mathcal{S}}$. This model is based on [9]. The encoder, f_e, transforms the input partial shape into a latent representation, $z \in \mathbb{R}^{n_z}$, of the body pose and human shape. The decoder, f_d, uses this latent code to deform a template body mesh, $\mathcal{S}_T$, into the pose and shape of the input. The result is a first estimation, $\hat{\mathcal{S}}_0$, of the ground-truth complete shape, $\mathcal{S}$.

To refine the first shape estimate, $\hat{\mathcal{S}}_0$, the corresponding latent code $\hat{z}_0$ is adjusted such that the decoded shape better fits the input partial shape. This is cast as the optimisation problem

$$\hat{\mathcal{S}} = \arg\min_{z} d_{\text{Chamfer}}(\mathcal{S}_p, f_d(z)). \tag{1}$$

The decoder, f_d, is taken as a black-box function. The objective function is the directed Chamfer distance [7] from the partial shape, $\mathcal{S}_p$, to the estimation, $\hat{\mathcal{S}}$. The directed Chamfer distance is important to only fit the partial shape where there is information and prevent uncontrolled deformations in holes (as reported in the experiments in Section 4.1). Additionally, the fitting makes use a higher-resolution shape than the one used in the encoder-decoder to fit clothing shape details more precisely.

Architecture The encoder-decoder is parametrised by a deep neural network with 3D convolutions. The encoder, f_e, follows a PointNet [20] architecture. It takes as input the set of mesh vertices, $\mathbf{V} \in \mathbb{R}^{n_v \times 3}$, and applies successive point convolutions. A point convolution consists in a shared multi-layer perceptron (MLP) applied on the features of a set of points, to produce transformed features of a possibly different size. The feature size of the successive layers are $(3, 64, 128, 1024)$. The last layer is max-pooled across the points into a vector, z_0, of size $n_z = 1024$. This vector is then refined using two densely connected layers of size 1024 to produce the latent vector z of size n_z as well. The decoder, f_d, concatenates the latent code, z, to each vertex of the template mesh and then applies a series of point convolutions, as above. The feature size of the layers in the decoder are $(3 + 1024, 513, 256, 128, 3)$.

Training The training strategy of the network follows [9]. The mesh of the SMPL body model [16] is used as the body mesh template, $\mathcal{S}_T$. The encoder-decoder network, f, is trained with supervision by learning to reconstruct the SMPL body model in randomly generated poses and shapes. The input training data is augmented with random subsampling of the points and random shifts in the positions. This makes the network robust to partial and irregular sampling and variations in the input data. The loss is the mean-squared error (MSE) on the point positions.

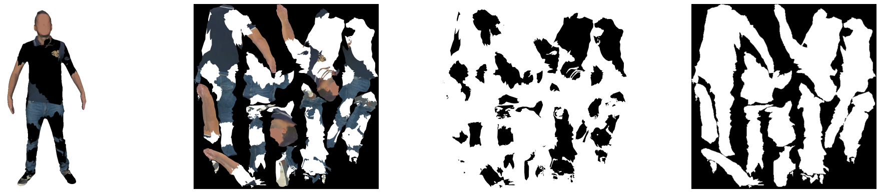
3.2 Body Texture Completion

After estimating the complete 3D shape (Section 3.1), a corresponding complete texture image is estimated with the following steps. First, the input partial texture is transferred onto the texture image of the estimated shape. Then, the regions to be inpainted are identified. Finally, the completion of the texture is performed by inpainting the partial texture image with specific constraints for handling the topology of the texture atlas. These steps are detailed below.

Partial texture transfer The method of Section 3.1 estimates a complete mesh $\hat{\mathcal{S}}$ aligned with the input partial shape $\hat{\mathcal{S}}_p$. The partial texture T_p of the input mesh is transferred onto $\hat{\mathcal{S}}$ by a ray-casting algorithm that propagates the texture information along the normal directions. The result is a mesh with a complete shape and a partial texture, $\hat{\mathcal{X}}_{T_p} = (\hat{\mathcal{S}}, \hat{T}_p)$. This is illustrated by the mesh in Fig. 5a, where the regions without mapped colour are rendered in black (default background colour). These regions must be identified on the corresponding texture image prior to inpainting.

Identification of the regions with missing texture If no texture information is transferred in a particular region of the mesh $\hat{\mathcal{X}}_{T_p}$, the corresponding region in the texture image is left unmodified with the default black background colour. Thus, the black pixels inside the charts of the texture atlas indicate missing texture information. This is illustrated in Fig. 5b where the identified regions without texture are highlighted in white. A binary mask M with the same dimension as the partial texture image is derived such that $M(i, j) = 0$ if the pixel (i, j) corresponds to a missing texture information due to partial data, and $M(i, j) = 1$ otherwise. An example of the computed mask is shown in Fig. 5c. The corresponding background mask M_b is shown in Fig. 5d.

Texture inpainting To the transferred partial texture image $\hat{T}_p$ is associated the mask M indicating the regions with missing information. Additionally, the background mask M_b of the texture is known from the definition of the charts



(a) Partial texture transferred on a complete mesh.

(b) Identified regions with missing texture (white).

(c) Binary mask of the partial texture.

(d) Binary background mask.

Fig. 5: Identification of missing regions on a partial texture and binary masks calculation.

in the texture atlas (Section 2). Given this information, the problem of texture completion is turned into an image inpainting task. However, the inpainting should only occur in the foreground regions, *i.e.* on the charts of the texture atlas. Moreover, the non-informative background of the image must be explicitly ignored to prevent irrelevant background colour (*e.g.* black) to propagate onto the charts. The proposed adapted inpainting algorithm is detailed below.

Image inpainting is extensively studied in the literature [29, 31, 30, 15]. While some works focus on image inpainting with regular masking shapes (*e.g.* rectangular masks) [30, 29], more recent works try to address the inpainting problem in case of irregular masking shapes [15, 31]. In our case, the masks of the missing texture are derived from partial 3D shapes. This makes the masks irregular and not restricted to specific shapes as it can be observed in Fig. 5c. Consequently, the selected image inpainting approach should take into account these irregularities. Accordingly, we build on the method proposed in [15] handling irregular masks. In [15], the authors use partial convolutional layers instead of conventional convolutional layers. These layers consist of mask-aware convolutions and a mask update step.

Given a binary mask M, partial convolutions extend standard convolutions to focus the computations on the information from unmasked regions (*i.e.* pixels (i, j) such that $M(i, j) = 1$) and discard the information from masked regions. With the goal of inpainting, the masks are updated after every partial convolutional operation by removing the masking (*i.e.* changing the mask value from 0 to 1) for each location that was involved in the convolution. The mask update forces the masked regions to disappear after a sufficient number of updates. More formally, let W be the weights for a specific convolution filter and b its corresponding bias. T_w are the feature values for the current sliding window and M is the corresponding binary mask. The partial convolution at at every location, similarly defined in [15], is expressed as:

$$t_c = \begin{cases} W^T(T_w \odot M) \cdot \frac{\mathrm{sum}(\mathbf{1})}{\mathrm{sum}(M)} + b & \text{if } \mathrm{sum}(M) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\odot$ denotes element-wise multiplication, and $\mathbf{1}$ has same shape as M but with all elements being 1. After every partial convolution, a masked value in M $(M(i,j) = 0)$ is updated to unmasked $(M(i,j) = 1)$ if the convolution was able to condition its output on at least one valid input value. In practice this is achieved by applying fixed convolutions, with the same kernel size as the partial convolution operation, but with weights identically set to 1 and no bias.

One important observation in the two texture atlases provided in Fig. 2, is that they contain some non-informative black regions used as background to gather the body charts in a single image. The inpainting of the missing texture information (white regions in Fig. 5b) could be impacted by the non-informative background (*i.e.* black) using the original form of partial convolutions introduced in [15]. This is confirmed and visualised by experiments in Section 4.3. As a solution, we propose to ignore these regions during the partial convolutions as done with the masked values of the missing texture to be recovered. However, these regions should not be updated during the mask update as the background mask should stay fixed through all the partial convolution layers. This is achieved by including the *background mask* M_b of the texture image in the partial convolution as follows,

$$t_c = \begin{cases} W^T (T_w \odot M \odot M_b) \cdot \frac{\mathrm{sum}(\mathbf{1})}{\mathrm{sum}(M \odot M_b)} + b & \text{if } \mathrm{sum}(M \odot M_b) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The background mask M_b is passed to all partial convolutions layers without being updated by applying *do-nothing* convolution kernels with the same shape as the ones used for the masks M. A *do-nothing* kernel consists of a kernel with zeros values everywhere except for the central location which is set to 1. Moreover, before updating the original mask M we apply on it this background mask M_b by element-wise multiplication so that we guarantee that the mask M will not be updated using the background regions.

The aforementioned partial convolutional layers are employed in a UNet-like architecture [21] instead of standard convolutions. Several loss functions are used to optimise the network. Two pixel-wise reconstruction losses are defined separately on the masked and unmasked regions with a focus on masked regions. Style transfer losses are also considered by constraining the feature maps of the predictions and their auto-correlations to be close those of the ground truth [8]. Finally a Total-Variation (TV) loss [11] is employed on the masked regions to enforce their smoothness. For more details about the aforementioned inpainting method are presented in [15].

The inpainting task is facilitated by the fact that the texture atlas images to inpaint have a fixed arrangement of the charts. This is because they are all defined on the same template mesh $\mathcal{X}_\mathcal{T}$ of used for the shape completion in Section 3.1. This means that semantic body regions are placed consistently on the texture images, regularising the inpainting problem.

In the experiments (Section 4.3), two training strategies are investigated for this inpainting method. First, the network is trained from scratch on the

3DBodyTex.v2 dataset [25]. Second, the network, pretrained on the ImageNet dataset [5], is fine-tuned on the 3DBodyTex.v2 dataset [25].

4 Experiments

This work focuses on the completion of textured 3D human shapes using the 3DBodyTex.v2 dataset introduced in the SHARP challenge [25].

The ***SHA**pe **R**ecovery from **P**artial Textured 3D Scans* (SHARP) [25] challenge aims at advancing the research on the completion of partial textured 3D shape. Two challenges are proposed with two corresponding datasets of 3D scans: 3DBodyTex.v2, a dataset of human scans, and 3DObjectTex, a dataset of generic objects. 3DBodyTex.v2 contains about 2500 humans scans of a few hundred people in varied poses and clothing types. It is an extension of 3DBodyTex [22].

4.1 Shape completion

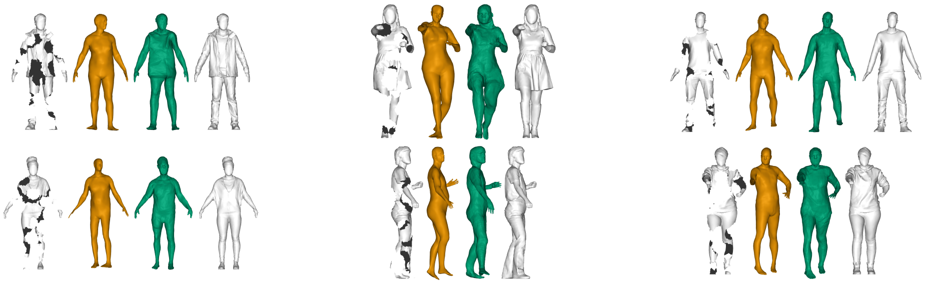


Fig. 6: Results of shape completion for 6 examples of the test set. From left to right: input partial shape (white), initial shape estimate (orange), refined shape estimate (green), ground truth (white). In the input partial shapes, the visible interior surface is rendered in black.

Fig. 6 shows the results of the shape completion (Section 3.1). It can be seen that initial shape estimate (orange) captures the pose of the partial input but not the loose-fitting clothing. The refined shape (green) represents the clothing more accurately. This shows the validity of the approach in recovering clothed body shapes.

However, several limitations are observed. A topology significantly different from the template body mesh is difficulty recovered. This happens for example with hair and clenched fists. Moreover, in the example in row 1 column 3 of Fig. 6, the left foot is not recovered because it is completely cropped from the partial input. This suggests that the shape estimation can fail locally on an extremity of the body when no information is available in the partial shape. This could be improved by retraining the encoder-decoder model on a dataset of partial shapes. Furthermore, the pose of the human skeleton is not always sufficiently accurate in the first shape estimate produced by the encoder-decoder network. As a consequence, the refinement fails and the final shape estimate is not correctly aligned. This is due in part to the variety in clothing shapes for which the network has not been trained. Similarly, in row 1 column 2 of Fig. 6, the refined shape (green) does not capture the shape of the skirt realistically. This is also due to the network being trained on body shapes only. The encoder-decoder, f, could thus profit from a training or fine-tuning on a dataset of clothed shapes and also possibly from adaptations of the architecture to handle the more complex deformations of the clothing.

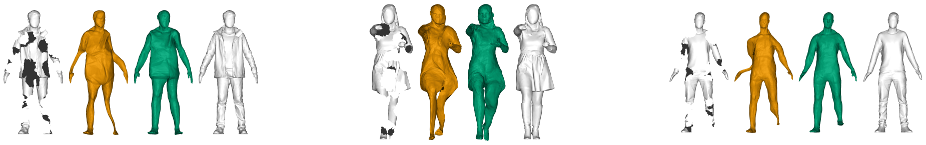


Fig. 7: Shape refinement with symmetric versus directed (one-way) Chamfer distance for three examples of the test set. The one-way Chamfer distance (green) is the one retained in the proposed approach. From left to right: input partial shape (white), refined shape with symmetric Chamfer distance (orange), refined shape with directed (one-way) Chamfer distance (green), ground truth (white). In the input partial shapes, the visible interior surface is rendered in black.

Fig. 7 illustrates the importance of the chosen objective function in the optimisation problem (1) to refine the initial shape estimate. With a symmetric Chamfer distance (from partial input to estimated shape and conversely), the shape refinement fails (orange shape in Fig. 7). With a directed Chamfer distance (from partial input to estimated shape only), the shape refinement is sound. This is due to the holes in the input partial shape. With the symmetric distance, the measure from the estimated shape to the partial input has the effect of dragging the estimated shape into regions of the partial input without holes. This creates unrealistic distortions in the estimated shape.

4.2 Texture transfer

Fig. 8 illustrates the texture transfer from the partial shape onto the refined shape estimate. Overall, the texture is mapped correctly when the shape estimate is close the partial shape. When the estimated shape is incorrect (e.g. foot in third column), the transferred texture is directly impacted.

4.3 Texture inpainting

Visual results on the validation set for texture completion are presented in Fig. 9. The completed textures are displayed on the completed shapes obtained with

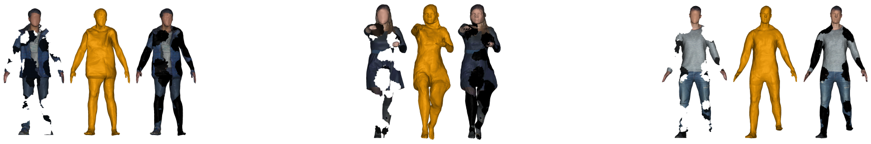
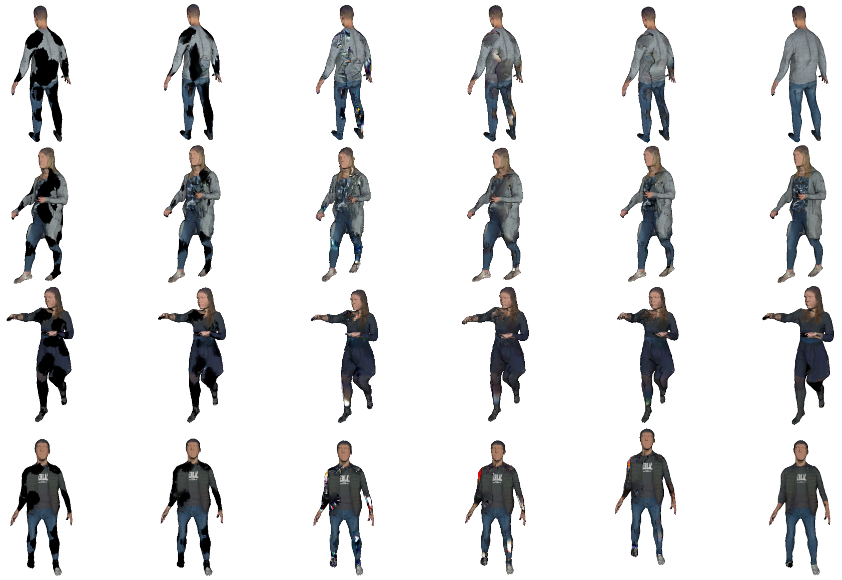


Fig. 8: Illustration of the texture transfer (right) from the partial shape (left) onto the refined shape estimate (orange, middle) for three examples of the test set.



(a) Input partial (b) Pretrained (c) Pretrained + M_b (d) Scratch + M_b (e) Fine-tuning + M_b (f) Ground truth

Fig. 9: Results for texture inpainting using different strategies.

the method from Section 4.1. The first column (a) shows the input partial texture while the last column (f) depicts the ground-truth complete texture. The intermediate columns show the inpainting results and training strategies: (b) the original inpainting model, pretrained on ImageNet; (c) the improved inpainting model with the proposed background masks (Section 3.2), pretrained on ImageNet; (d) the improved inpainting model with background masks, trained from scratch on 3DBodyTex.v2 and (e) the improved inpainting model with background masks, pretrained on ImageNet and fine-tuned on 3DBodyTex.v2.

As seen in Fig. 9b, the original inpainting model pretrained on ImageNet is able to complete some of the missing regions with colour matching the local context. However, the holes are not completed fully. This is due to the black background of a texture atlas which is used as a local context for inpainting, as explained in Section 3.2. With the addition of the proposed background masks to the inpainting model (Fig. 9c), the holes are fully completed. This validates the proposed approach tailored to the data at hand. However, the completed colour does not match the local context accurately and contains artefacts. For example, some holes are filled in with random colour patterns and some white patches are produced in dark regions. Training the improved inpainting model from scratch on 3DBodyTex.v2 (Fig. 9d) reduces the colour artefacts. The completed colour patterns are more regular but the colour does not follow closely the surrounding context. Thus, the 3DBodyTex.v2 dataset seems enough to regularise the colour pattern inside the clothing but not rich and varied enough for the model to learn what colour to complete with. Indeed, 3DBodyTex.v2 is relatively small (a few thousands samples) in comparison to ImageNet (millions of examples). Finally, the proposed method of fine-tuning on 3DBodyTex.v2 the improved inpainting model pretrained on ImageNet (Fig. 9e) gives the best results by producing regular colour patterns matching the local surrounding regions closely.

5 Conclusion

This work proposes 3DBooSTeR, a novel approach for the completion of partial textured human meshes. The tasks of 3D shape completion and texture completion are addressed sequentially. First, the 3D body shape completion is performed with an encoder-decoder system that encodes a partial input point cloud of a human shape into a latent representation of the complete shape and decodes this representation into a completed shape by deforming a template body mesh. The estimated shape is further refined to better match the clothing by adjusting the encoded latent code with an optimisation procedure adapted to the input partial data. The partial texture information is then projected onto the estimated shape. The regions of the mesh to be completed are identified from the 3D shapes and mapped onto the texture image. The texture completion is then seen as a texture inpainting problem. For this task, a novel inpainting method tailored to texture maps is proposed. It is specifically designed to handle holes of irregular shape and to be robust to irrelevant background image information. Experiments on 3DBodyTex.v2 show the validity of the proposed approach on partial data. Future work includes making the encoder-decoder network capture the clothing shape better, by training on a dataset of clothing shape and/or adapting the architecture. The texture transfer could be made more robust to bad shape estimates by enforcing a continuity in the mapping from partial shape to estimated shape. Overall, the pipeline might profit from a coupling between the shape and texture completion, for example in an end-to-end neural network architecture. This involves at least designing differentiable alternatives to the two intermediate manual steps of shape refinement and texture transfer.

Acknowledgements

This work was supported by the Luxembourg National Research Fund (FNR), projects BODYFIT (11806282) and IDFORM (11643091), and Artec3D.

References

1. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM SIGGRAPH 2005 Papers, pp. 408–416 (2005)
2. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6970–6981 (2020)
3. Dai, A., Ruizhongtai Qi, C., Nießner, M.: Shape completion using 3d-encoder-predictor cnns and shape synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5868–5877 (2017)
4. Davis, J., Marschner, S.R., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission. pp. 428–441. IEEE (2002)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
6. 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)
7. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 605–613 (2017)
8. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
9. 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). pp. 230–246 (2018)
10. 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: Proceedings of the IEEE international conference on computer vision. pp. 85–93 (2017)

11. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
12. 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)
13. Li, C.L., Simon, T., Saragih, J., Póczos, B., Sheikh, Y.: Lbs autoencoder: Self-supervised fitting of articulated meshes to point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11967–11976 (2019)
14. Litany, O., Bronstein, A., Bronstein, M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1886–1895 (2018)
15. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: The European Conference on Computer Vision (ECCV) (2018)
16. 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)
17. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. ACM siggraph computer graphics **21**(4), 163–169 (1987)
18. Ma, Q., Yang, J., Ranjan, A., Pujades, S., Pons-Moll, G., Tang, S., Black, M.J.: Learning to dress 3d people in generative clothing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6469–6478 (2020)
19. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)
20. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
21. 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. pp. 234–241. Springer (2015)
22. Saint, A., Ahmed, E., Cherenkova, K., Gusev, G., Aouada, D., Ottersten, B., et al.: 3dbodytex: Textured 3d body dataset. In: 2018 International Conference on 3D Vision (3DV). pp. 495–504. IEEE (2018)
23. Saint, A., Cherenkova, K., Gusev, G., Aouada, D., Ottersten, B., et al.: Bodyfitr: Robust automatic 3d human body fitting. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 484–488. IEEE (2019)
24. Saint, A., Shabayek, A.E.R., Aouada, D., Ottersten, B., Cherenkova, K., Gusev, G.: Towards automatic human body model fitting to a 3d scan. In: Proceedings of 3DBODY. TECH 2017-8th International Conference and Exhibition on 3D Body Scanning and Processing Technologies, Montreal QC, Canada, 11-12 Oct. 2017. pp. 274–280. Hometrica Consulting (2017)
25. SHARP team: SHARP Challenge ECCV 2020. http://gitlab.uni.lu/cvi2/eccv2020-sharp-workshop (2020), online; accessed 02/08/2020
26. Szeliski, R., Lavallée, S.: Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. International journal of computer vision **18**(2), 171–186 (1996)

27. Verma, N., Boyer, E., Verbeek, J.: Feastnet: Feature-steered graph convolutions for 3d shape analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2598–2606 (2018)
28. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
29. Yeh, R.A., Chen, C., Yian Lim, T., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with deep generative models. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5485–5493 (2017)
30. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018)
31. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4471–4480 (2019)