UV-based reconstruction of 3D garments from a single RGB image

Albert Rial-Farràs[1,2], Meysam Madadi[3], and Sergio Escalera[2,3]
[1] Facultat d'Informàtica de Barcelona (FIB), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
[2] Facultat de Matemàtiques i Informàtica, Universitat de Barcelona (UB), Barcelona, Spain
[3] Computer Vision Center (CVC), Barcelona, Spain

***Abstract*— Garments are highly detailed and dynamic objects made up of particles that interact with each other and with other objects, making the task of 2D to 3D garment reconstruction extremely challenging. Therefore, having a lightweight 3D representation capable of modelling fine details is of great importance. This work presents a deep learning framework based on Generative Adversarial Networks (GANs) to reconstruct 3D garment models from a single RGB image. It has the peculiarity of using UV maps to represent 3D data, a lightweight representation capable of dealing with high-resolution details and wrinkles. With this model and kind of 3D representation, we achieve state-of-the-art results on the CLOTH3D++ dataset, generating good quality and realistic garment reconstructions regardless of the garment topology and shape, human pose, occlusions and lightning.**

I. INTRODUCTION

Inferring 3D shapes from a single viewpoint is an essential human vision feature extremely difficult for computer vision machines. For this reason, there has always been a great deal of research devoted to building 3D reconstruction models capable of inferring the 3D geometry and structure of objects or scenes from a single or multiple 2D pictures. Tasks in this field go from reconstructing objects like cars or chairs to modelling entire cities. These models can be used for a wide variety of applications such as 3D printing, simulation of buildings in the civil engineering domain, or generation of artificial objects, humans and scenes for videogames and movies.

One field in which the research community and industry have been working for years has been the study of human dynamics. Accurately tracking, capturing, reconstructing and animating the human body, face and garments in 3D are critical tasks for human-computer interaction, gaming, special effects and virtual reality. Despite the advances in the field, most research has concentrated only on reconstructing unclothed bodies and faces, but modelling and recovering garments have remained notoriously tricky.

For this reason, our work pushes the research on the specific domain of garments, learning clothing dynamics and reconstructing clothed humans. This topic has many potential applications and benefits: allow virtual try-on experiences when buying clothes, reduce designers and animators workload when creating avatars for games and movies, etc.

First-generation methods that tried to recover the lost dimension from just 2D images were concentrated on understanding and formalising, mathematically, the 3D to 2D projection process [20], [27]. However, this kind of solutions required multiple images captured with well-calibrated cameras and accurately segmented, which in many cases is not possible or practical.

Recent advances in deep learning algorithms, as well as the increasing availability of large training datasets, have resulted in a new generation of models that can recover 3D models from one or multiple RGB images without the complex camera calibration process. In this project, we are interested in these advantages, so we focus on developing a solution of this second category, using and testing different deep learning techniques. Specifically, we are interested in UV maps compared to other 3D surface representations such as meshes, point clouds or voxels, which are the ones commonly used in other 3D deep learning models [11], [53], [43], [49], [54], [28], [40], [41], [13], [33], [51], [52], [18], [19]. UV maps allow us to use standard computer vision architectures, usually intended for images and two-dimensional inputs/outputs. In addition, and even more important, UV maps are lightweight and capable of modelling fine details, a feature necessary for modelling challenging dynamic systems like garments.

In this work, we present a model based on Generative Adversarial Networks (GANs) [17] that has the peculiarity to use UV maps to represent 3D data. It is based on LGGAN (Local Class-Specific and Global Image-Level Generative Adversarial Networks) [48], used previously on semantic-guided scene generation tasks like cross-view image translation and semantic image synthesis and characterised by combining two levels of generation, global image-level and local class-specific.

Our contributions are summarised below:

1) We adapt the LGGAN architecture to predict garment UV map from the input image. We also introduce 3D loss functions to improve the surface quality.
2) In addition, this work also studies the feasibility and impact of using UV map representations for 3D reconstruction tasks, demonstrating that it is an interesting option that could be applied in many other 3D tasks.
3) As a result, in this work we propose a system that can adequately learn garment dynamics and reconstruct clothed humans from a single RGB image, using UV maps to represent 3D data and achieving state-of-the-art results in CLOTH3D++ [30] dataset.

II. RELATED WORK

In this section, we review the state-of-the-art on 3D object reconstruction and more specifically the garment reconstruc-

tion field. We also focus on the 3D representations used in these works and the differences between them.

The first works approached the problem from a geometric perspective and tried to understand and formalise the 3D projection process mathematically [20], [46], [9], [27]. Although these solutions are very successful in some scenarios, they are subject to several restrictions. They require a great number of images from different viewpoints of the object and this cannot be non-lambertian (e.g. reflective or transparent) nor textureless. They also require the objects to be accurately segmented from the background and well-calibrated cameras, which is not suitable in many applications.

All these restrictions lead the researchers to draw on learning-based approaches, that consider single or multiple images and rely on the shape prior knowledge learnt from previously seen objects. With the success of deep learning architectures, and more importantly, the release of large-scale 3D datasets, learning-based approaches have achieved great progress and very promising results.

Unlike 2D images, which are always represented by regular grids of pixels, 3D shapes have various possible representations, being voxels, point clouds and meshes the most common. This has led to 3D reconstruction models that are very different from each other, as their architectures largely depend on the representation used.

a) Voxel-based representations: Voxel-based were among the first representations used in deep learning techniques to reconstruct 3D objects. One voxel is basically a 3D base cubical unit that, together with other voxels, form a 3D object inside a regular grid in the 3D space. It can be seen as a 3D extension of the concept of pixel. Due to this 2D analogy with pixels, a data type used in computer vision for decades, in the last few years many different approaches dealing with voxels have appeared [11], [53], [49], [54]. A problem with this representation is that the results may lack detail since the resolution is limited by the amount and size of voxels, so to have high quality results a high memory consumption is required.

b) Point-based: 3D point clouds are another alternative to represent 3D objects that have been widely used in several works [13], [33]. A point cloud is a set of unstructured data points in a three-dimensional coordinate system that approximates the geometry of 3D objects. Unlike voxels, this representation is simple and efficient, but the connectivity absence among points results in an ambiguity about surface information. Moreover, the lack of structure and order they present makes networks using this type of representation hard to train.

c) Mesh-based: Meshes are one of the most popular representations used for modelling 3D objects. It is a geometric data structure that allows the representation of surface subdivisions by a set of polygons, faces, which are made up of vertices. The vertices describe how the mesh coordinates x, y, z exist in the 3D space and connect to each other forming the faces. There have been a lot of works using this type of representation [51], [52], [18]. Availability of topology in meshes makes them a good candidate to describe local regions on surfaces. Graph Convolutional Networks (GCN) are common deep learning algorithms to process 3D meshes.

d) UV-based: Finally, there are a few novel works that also use UV maps to represent 3D data [10], [39], [14] and reconstruct and model objects, faces and hands. A UV map is the flat representation of the surface of a 3D model, usually used to wrap textures easily and efficiently. In the works mentioned before, UV maps are used to store in a 2D flat surface the 3D coordinates of the vertices/points of the objects. Having this 2D representation allow the models to predict geometry in an image-to-image translation fashion.

A. Garment reconstruction

Inferring 3D cloth models from single images is an extremely challenging task comparing to simple and static objects like chairs or cars. Garments are not only very dynamic objects but also belong to a domain where there is a huge variety of topologies and types. Early works approached the problem from a geometric perspective [8], [56], [22]. Then, the advent of deep learning achieved impressive progress in the task of reconstructing unclothed human shape and pose [7], [23], [25], [26], [37], [38], [47], [55]. Nevertheless, recovering clothed human 3D models did not progress in the same way, not only because of the challenges mentioned before but also as a consequence of the lack of large annotated datasets. It is only recently, in the last few years, that new large datasets and deep learning works focused on this domain have started to emerge, trying to push its state-of-the-art.

In the specific field of garment reconstruction, frameworks also use different representations for clothing. There are works that use voxel-based representations [50] or visual hulls [35], although they do not allow high resolution. There are also some works that use implicit functions [44], [45], [12], but their limitation is that the output does not have an explicit model to control its pose and shape. Other works follow a quite standard practice in garment reconstruction which is to represent garments with meshes but as an offset over the Skinned Multi-Person Linear model (SMPL) body mesh [6], [1], which is a skinned vertex-based model that accurately represents a wide variety of body shapes in natural human poses [29]. Finally, to our best knowledge, there is just a single work [2] that also uses UV maps for the task of 2D to 3D garment reconstruction. In their model, authors also follow the standard practice mentioned before and use displacement UV maps, which basically contain 3D vectors that displace the underlying surface and that are defined on top of the SMPL body. These representations defined as an offset over the SMPL body have the disadvantage that it may fail for loose-fitting garments, which have significant displacements over the shape of the body. In [2] they do not tackle this issue and fail for some types of garments such as dresses. Our approach, using also displacement UV maps, is able to handle loose-fitting garments by using a semantic-guided class-specific generation network that learns specific features for each type of garment within a unique system.

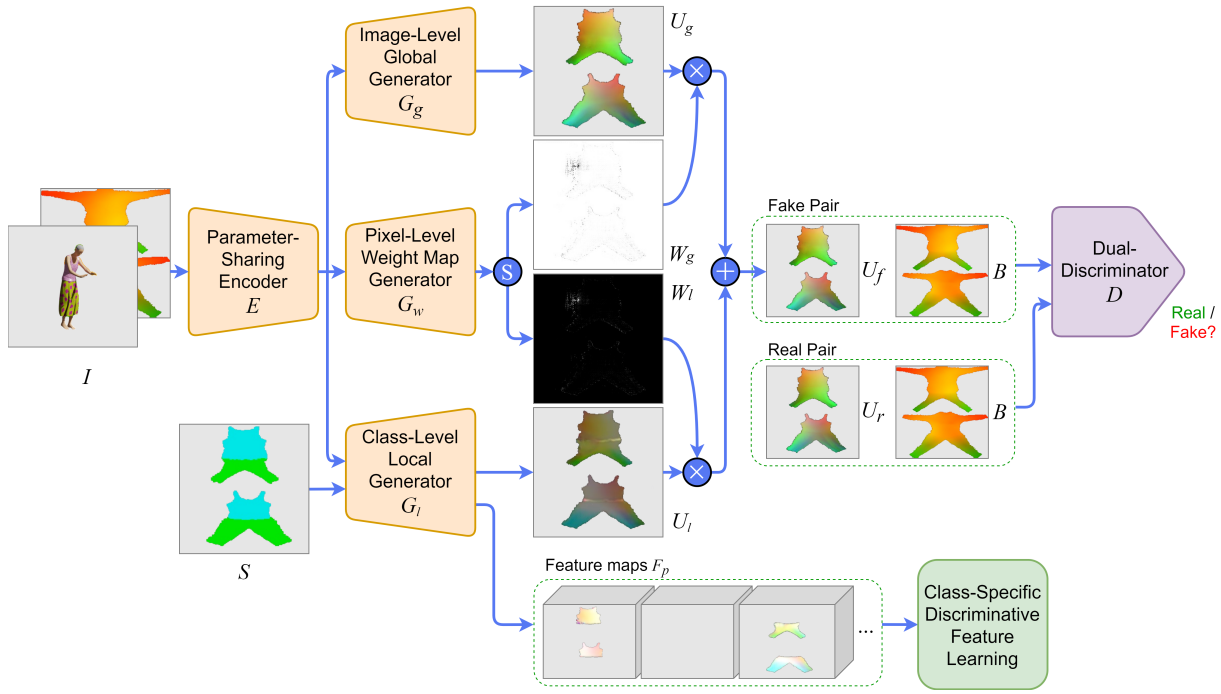


Fig. 1: Overview of the proposed LGGAN. The symbol $\oplus$ denotes element-wise addition, $\otimes$ element-wise multiplication and (s) channel-wise softmax. Source: Own elaboration adapted from [48].

III. METHODOLOGY

Our goal is to reconstruct 3D clothed humans from just a single 2D image. To do so, we use displacement UV maps to represent 3D data and approach the task of 2D to 3D reconstruction as an image-to-image translation problem, in which the RGB input image is translated into a UV map. In this problem, models learn mappings from input images to output images. However, as in our case input and output do not have a clear pixel correspondence between them, as pixels follow a very different path from RGB image (input) to the UV map (output), we tackle the problem as done in cross-view image translation task [42]. In this task, models use a semantic guide that helps them to learn the correspondences between the target and source images. Following a similar approach, we adapt a state-of-the-art model in this task, LGGAN (Local class-specific and Global image-level Generative Adversarial Networks) [48], and study its performance in the UV map domain. This model is based on GANs [17], more specifically in CGANs [34], and is mainly composed of three parts/branches: a semantic-guided class-specific generator modelling local context, an image-level generator modelling the global layout, and a weight-map generator for joining the local and the global generators. An overview of the proposed system is shown in Fig. 1.

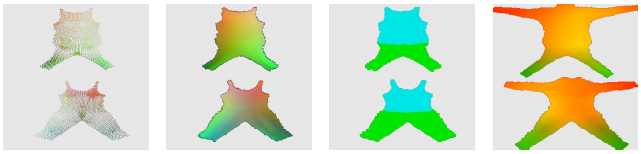
A. UV maps

We create UV maps B, S and U for the body mesh, garment semantic segments and garment mesh, respectively. For the body we use SMPL mesh and topology, following the shape from CLOTH3D++ [30]. To create the garment UV maps, we first align all the garment topologies with a canonical topology to have a homogeneous space. The reason is that each garment has a different topology which is known during training, but unknown at inference time. Therefore, to avoid predicting the garment topology at inference, we use a homogeneous topology, same as the body. To do so, we register garments on top of SMPL body using non-rigid ICP [3] as a preprocessing step similar to [5]. As in [2], we use displacement UV maps that store garment vertices as an offset over the estimated SMPL body vertices. Additionally, we create garment UV map of semantic labels, meaning each UV coordinate is assigned a color w.r.t. its garment class. An example of body and garment UV maps is shown in Fig. 2.

The UV coordinates are discrete since a 3D mesh is a discrete representation and has a finite number of vertices. Therefore, the UV maps have empty gaps between vertices, as can be seen in Fig. 2a. To avoid these gaps and make the UV map image smoother, we use image inpainting techniques to estimate the values of the empty spaces. Image inpainting is a form of image restoration that is usually used to restore old and degraded photos or repair images with missing areas. In particular, we use the Navier-Stokes based method [4], which propagates the smoothness of the image via partial differential equations and at the same time preserves the edges at the boundary [32]. With this technique, we obtain a UV map like the one in Fig. 2b.

B. LGGAN for 3D garment reconstruction

Like the original GAN [17], LGGAN is composed of a generator G and a discriminator D. The generator G consists of a parameter-sharing encoder E, an image-level global generator G_g, a class-level local generator G_l and a weight map generator G_w. The encoder E shares parameters to all the three generation branches to make a compact backbone



(a) Original UV (b) Inpainted UV (c) Semantic map (d) SMPL body map

Fig. 2: Examples of an original UV map, its inpainted version, a semantic segmentation map (each color represents a different garment type) and a SMPL body UV map.

network. Gradients from all the three generators G_g, G_l and G_w contribute together to the learning of the encoder. LGGAN, originally, is applied on semantic guided cross-view image generation in which the input RGB image is concatenated with a semantic map and fed to the backbone encoder E. This semantic map is a semantic segmentation of the target image, with a class label for every type of object appearing on it. In this paper, we adapt LGGAN to translate the input image to the garment UV map.

Our model expects an RGB image as input along with a conditioning on 1) UV map segmentation S that guides and controls it through the class-level local generator G_l, and 2) the SMPL body UV map B of the human to condition the image-level global generator G_g and the discriminator D. Note that at inference time the semantic map S is estimated by the same LGGAN model and the SMPL body surface, which is used to create body UV map B, is estimated by SMPLR [31].

LGGAN extends the standard GAN discriminator to a cross-domain structure that receives as input two pairs, each one containing a UV map and the condition of the model. Nevertheless, the goal of the discriminator D is the same, try to distinguish the generated UV maps from the real ones.

1) Parameter-Sharing Encoder: The first module of the network is a backbone encoder E. This module basically takes the input I and applies several convolutions to encode it and obtain a latent representation $E(I)$. It is composed of three convolutional blocks and nine residual blocks (ResBlocks) [21]. Note that the input I, as explained before, is the concatenation of the input RGB image and the SMPL body UV map B.

2) Class-Specific Local Generation Network: is a novel local class-specific generation network G_l that separately constructs a generator for each semantic class. It helps to have more diverse generation for different garment classes. Each sub-generation branch has independent parameters and concentrates on a specific class, thus, producing better generation quality for each garment and yielding richer local details. This means class-Specific Local Generation Network avoids converging to average dynamics among different garments. Its overview is shown in Fig. 3.

This generator receives as input the output of the encoder, the encoded features $E(I)$, which are fed into two consecutive deconvolutional blocks to increase the spatial size. The

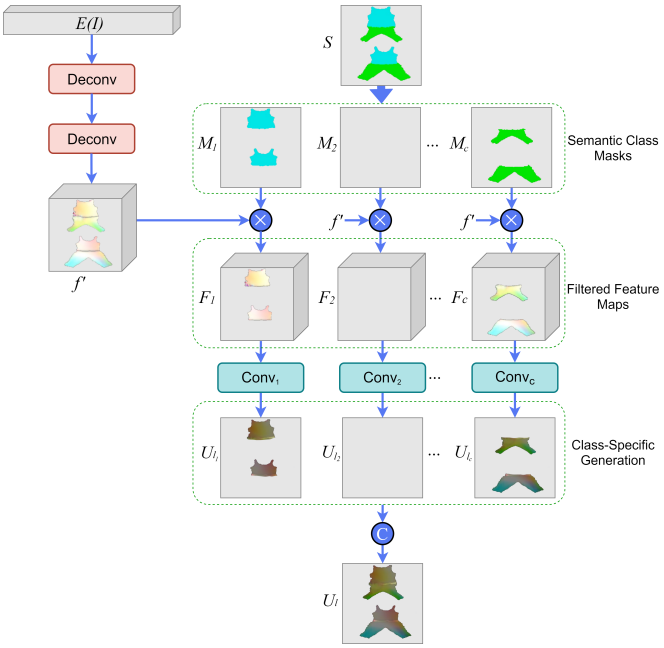


Fig. 3: Class-Specific Local Generation Network overview. The symbol $\otimes$ denotes element-wise multiplication, and Ⓒ channel-wise concatenation. Source: Own elaboration adapted from [48].

scaled feature map f' is then multiplied by the semantic mask of each class M_i, obtained from the semantic map S. By doing this multiplication, we obtain a filtered class-specific feature map for each one of the classes. This mask-guided feature filtering can be expressed as:

$$F_i = M_i \cdot f', \quad i = 1, 2, \ldots, c \tag{1}$$

where c is the number of semantic classes (i.e. number of types of garment). After computing these filtered feature maps, each feature map F_i is fed into a different convolutional block, the one for the corresponding class i, which generates a class-specific local UV map U_{l_i}. The loss used in this step is a semantic-mask guided pixel-wise L1 reconstruction loss:

$$\mathcal{L}_{\text{L1}}^{\text{local}} = \sum_{i=1}^{c} \mathbb{E}_{U_r, U_{l_i}} \left[\| U_r \cdot M_i - U_{l_i} \|_1 \right] \tag{2}$$

Finally, to generate the final output U_l of the local generator, an element-wise addition of all the class-specific outputs is applied:

$$U_l^L = U_{l_1} \oplus U_{l_2} \oplus \cdots \oplus U_{l_c} \tag{3}$$

3) Class-Specific Discriminative Feature Learning: To have more diverse generation for the different semantic classes, the authors of the original LGGAN also propose a classification-based feature learning module to learn more discriminative class-specific feature representations. This module receives as input a pack of feature maps produced from the different local generation branches $F_p = \{F_1, ..., F_c\}$ and predicts the classification probability of

the c classes of the image. As in the input image do not appear all garment types/classes, features from local branches corresponding to the void classes should not contribute to the classification loss. For this reason, the loss defined here is a Cross-Entropy (CE) loss, but filtering out the void classes by multiplying them with a void class indicator for each input sample. The indicator is a one hot vector $H = \{H_i\}_{i=1}^{c}$, with $H_i = 1$ for a valid class and $H_i = 0$ for a void one. The final CE loss is as follows:

$$\mathcal{L}_{\text{CE}} = -\sum_{m=1}^{c} H_m \sum_{i=1}^{c} 1\{Y(i) = i\} \log\left(f\left(F_i\right)\right) \quad (4)$$

being $1\{\cdot\}$ an indicator function that will return 1 if $Y(i) = i$, 0 otherwise, $f(\cdot)$ the function that produces the predicted classification probability given an input feature map $F(i)$, and Y the label set of all the classes.

4) Image-Level Global Generation Network: Apart from the local generator, LGGAN contains a global generation network G_g that captures global structure information or layout of the target images, in our case, UV maps. As in the local generator, this module receives as input the encoded features $E(I)$. The global result U_g is obtained through a feed-forward computation $U_g = G_g\left(E\left(I\right)\right)$.

5) Pixel-Level Fusion Weight-Map Generation Network: To combine local and global generated outputs, U_l and U_g, the LGGAN contains a pixel-level weight map generator G_w, which generates pixel-wise weights. Thus, the final output, a two-channel weight map W_f, is calculated as:

$$W_f = \text{Softmax}\left(G_w\left(E\left(S\right)\right)\right) \quad (5)$$

Finally, W_f is split to have a weight map W_l for the local generation and another one W_g for the global. The fused final generated UV map is computed as follows:

$$U_f = U_g \otimes W_g + U_l \otimes W_l \quad (6)$$

being $\otimes$ an element-wise multiplication operation.

6) Dual-Discriminator: The single domain vanilla discriminator from the original GAN [17] is extended to a cross-domain structure named semantic-guided discriminator. Its inputs are the SMPL body UV map B, the final output of the generator module (fake UV map) U_f and the ground truth output (real UV map) U_r.

$$\begin{aligned}\mathcal{L}_{\text{LGGAN}}\left(G, D\right) = \mathbb{E}_{B,U_r}\left[\log D\left(B, U_r\right)\right] \\ + \mathbb{E}_{B,U_f}\left[\log\left(1 - D\left(B, U_f\right)\right)\right]\end{aligned} \quad (7)$$

7) 3D loss functions: Apart from the losses of the original LGGAN model, described above, our model has four more types of 3D loss functions, which are applied on meshes. For this reason, to use them in our approach, in each step we recover 3D meshes by unwrapping the generated UV maps.

These losses are responsible for not only regressing the generated vertices to the ground truth but also ensuring that the generated mesh converges to a smooth and uniform shape. More specifically, two of the four losses are responsible for the direct comparison between predicted and ground truth vertices and face normals ($\mathcal{L}_{\text{smoothL1}}$, $\mathcal{L}_{\text{normal}}$), while the other two are in charge of adding regularisation to prevent the network of getting stuck into some local minimum ($\mathcal{L}_{\text{laplacian}}$, $\mathcal{L}_{\text{edge}}$).

Below we detail the contribution and computation of each of the losses. When formalising them, we use p for a vertex in the predicted mesh, q for a vertex in the ground truth mesh and $\mathcal{N}(p)$ for the set containing the neighbours of p.

a) Smooth L1 loss: This loss is a combination of L1 and L2 losses, used as a penalty to regress the predicted points p to its correct position, the ground truth q. It basically uses squared term (L2 term) if the absolute element-wise error falls below threshold β, and L1 term otherwise. This makes it less sensitive to outliers and, in some cases, prevents exploding gradients [15].

$$\mathcal{L}_{\text{smoothL1}} = \begin{cases} 0.5\left(p - q\right)^2/\beta, & \text{if } |p - q| < \beta \\ |p - q| - 0.5\beta, & \text{otherwise} \end{cases} \quad (8)$$

b) Surface normal loss: This term forces the normal of the faces from the predicted mesh to be consistent with the ground truth normals. Coming from Pixel2Mesh work [51], [52], this loss is defined as:

$$\mathcal{L}_{\text{normal}} = \sum_{p} \sum_{q=\arg\min_q\left(\|p-q\|_2^2\right)} \left\|(p - k)^T \cdot n_q\right\|_2^2 , \text{ s.t. } k \in \mathcal{N}(p) \quad (9)$$

being q the closest ground truth vertex to p found using chamfer distance, k a neighbour of p, and n_q the observed surface normal from ground truth at that vertex.

c) Laplacian smoothing regularisation: This loss, from a work by Nelan et al. [36], prevents the vertices from moving too freely, avoiding the output mesh from deforming too much and ensuring that it has a smooth surface.

$$\mathcal{L}_{\text{laplacian}} = \sum_{p} \sum_{k \in \mathcal{N}(p)} \frac{1}{|\mathcal{N}(p)|}(k - p) \quad (10)$$

d) Edge length regularization: To avoid having flying vertices, we penalise long edges by adding one last loss, extracted also from [51], [52].

$$\mathcal{L}_{\text{edge}} = \sum_{p} \sum_{k \in \mathcal{N}(p)} \|p - k\|_2^2 \quad (11)$$

Finally, the overall mesh loss is computed as a weighted sum of all these four losses, and contributes to all the three LGGAN generation branches, G_g, G_l and G_w, and the encoder E:

$$\mathcal{L}_{\text{mesh}} = \lambda_s \mathcal{L}_{\text{smoothL1}} + \lambda_n \mathcal{L}_{\text{normal}} + \lambda_l \mathcal{L}_{\text{laplacian}} + \lambda_e \mathcal{L}_{\text{edge}} \quad (12)$$

IV. EXPERIMENTS

A. Dataset

The dataset used for training and evaluating our model is CLOTH3D++ [30], which extends CLOTH3D [5], and was introduced in 2020 as the first large-scale synthetic dataset of 3D clothed human sequences. It has over 2 million 3D samples with a large variety of garment types, topology,

shape, size, tightness and fabric. Garments are simulated on top of thousands of different human pose sequences and body shapes, generating realistic cloth dynamics.

B. Evaluation metrics

To evaluate our approach and the different experiments done, we follow [30] and use their definition of the surface-to-surface (S2S) error metric, an extension of the chamfer distance (CD) that has the peculiarity to compute the distance based on the nearest faces rather than nearest vertices. As S2S evaluate the results based on mesh format, during evaluation we unwrap UV map representations into 3D meshes.

C. Training details

Training and experiments were performed on an NVIDIA GeForce GTX 1080 Ti GPU, with 11GB of memory and CUDA Version 11.0. RGB images are resized to 256×256 and aligned so the human is centered on it. The number of training epochs is set to 50, with a batch size of 4. Dropout is set to 0.5 and the weights of the 3D mesh losses defined before (Equation 12) are empirically set as: $\lambda_s = 1.0$, $\lambda_n = 0.01$, $\lambda_l = 0.5$ and $\lambda_e = 0.5$. The β parameter of the $\mathcal{L}_{\text{smoothL1}}$ loss is set to 1, as default (Equation 8).

Our LGGAN model is trained and optimised in an end-to-end fashion. We follow the optimisation method in [17] to optimise our LGGAN model, i.e. one gradient descent step on generators and discriminator alternately. We first train the encoder E and the three generators G_g, G_l and G_w with D fixed and then we train D with E, G_g, G_l and G_w fixed. The solver used is Adam [24] with momentum terms $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The initial learning rate used for Adam is 0.0001, but from epoch 25 onwards, we start to decrease it by $4\text{e}{-6}$. Finally, network weights are initialised with Xavier [16] strategy.

D. Ablation study

In this section, we present an ablation study of our model, analysing the contribution of each different technique designed, in a qualitative and quantitative manner. To do so, we take apart one component of our system at a time and show S2S error metric on the test set as well as some 3D reconstruction examples.

The quantitative study is shown in Table I. As we can observe, the full model is the one with the best performance, having an average S2S error of 19.1 mm. The parts that seem to contribute less to the performance of the model are the mesh losses and the conditioning on the SMPL body UV map, since the errors remain practically unchanged. On the other hand, the part that seems to contribute the most to the performance of the model is using displacement UV maps as our 3D representation. When we use original UV maps, which are not defined on top of SMPL, the performance in the test set decreases dramatically, having a much higher error in all types of clothing. The class-level local generator and the inpainting technique applied in the pre-processing of the UV maps are also key to our model, helping it to decrease its error significantly in most of the garment types.

TABLE I: Ablation study that evaluates the contribution of the different components of our model to its performance. Evaluate it computing S2S error (in mm) per garment type on the test set. Small is better (best values in bold).

Model	**S2S error (in mm)**						
	Top	T-shirt	Trousers	Jumpsuit	Skirt	Dress	All
Full model	**18.1**	**21.6**	**15.3**	**18.3**	25.6	**21.1**	**19.1**
-Mesh losses	18.3	21.9	15.4	18.5	**25.1**	**21.1**	19.2
-Condition on body	18.9	22.3	16.3	**18.3**	26.4	21.6	19.5
-Displacement maps	45.6	44.5	42.1	41.4	47.8	46.5	43.7
-Local generator	24.5	28.3	23.2	26.3	44.9	32.1	27.8
-Inpainting	20.9	24.1	22.6	21.3	42.6	31.2	24.9

In Fig. 4 we show a qualitative analysis of the 3D reconstructions of different types of garments when removing one component of our final system at a time, as done above. As we can see, the full model is the one that has more realism and has the best resemblance with the ground truth. Our model is capable of learning and modelling the dynamics of garments, regardless of the action, pose, race, gender and shape of the human. It is also able to accurately reconstruct garments regardless of the image point of view, recovering also the garment parts that are occluded. However, the reconstructions of our best model are not perfect at all. It is still not capable of generating garments as smooth as those of the ground truth, and, although it is able to reconstruct properly loose-fitting garments such as dresses or skirts thanks to the semantic-guided class-specific generation branch, it has some difficulties in modelling the larger displacements over the body.

In this analysis, we can appreciate much more the difference between the full model and the one not computing mesh losses. The generated reconstructions of the model without mesh losses are not as smooth as the reconstruction of the full model. Regarding conditioning on the SMPL body UV map, we observe that it has virtually no impact. We can also confirm that the major contribution comes from using displacement UV maps, as the model not using them has difficulties in capturing the human's movement and it has lots of artefacts. Lastly, we observe that when not using the inpainting technique or removing the local generator, reconstructions have also lots of lumps and wrinkles. Furthermore, the generated garments of the model without the local generator have quite significant artefacts at the edges where the front and the back of the garment (separated in the UV map) are joined.

E. Comparison with state-of-the-art methods

To the best of our knowledge, the best current state-of-the-art model in the task of RGB to 3D garment reconstruction for the CLOTH3D++ dataset is its baseline, presented in [30]. We do not only compare our model against it but also against SMPLicit [12], a novel generative model that can be used for 3D reconstructions of clothed humans and that has demonstrated very good results on other datasets, competing with or outperforming related works [2], [45].

In Table II we compare our model quantitatively against these two models using the S2S metric per garment type. As SMPLicit is not capable of inferring dresses and jumpsuits,

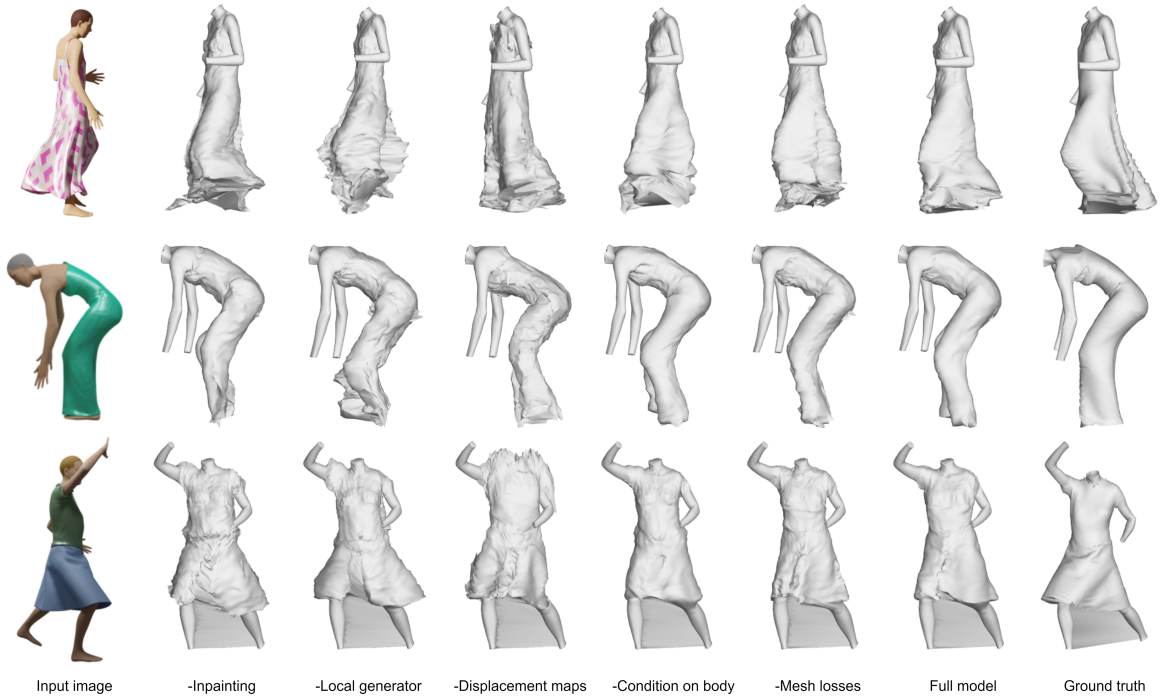


Fig. 4: Qualitative ablation study evaluating the contribution of the different components of the system to its performance.

due to how it was trained, we do not take these garment types into account when computing its metrics. It can be observed that our model obtains a lower error across all garment types, with an average error of about 12 mm less against CLOTH3D++ baseline and 24 mm less against SMPLicit.

TABLE II: S2S error (in mm) per garment type comparison against SOTA in CLOTH3D++ dataset. Small is better (best values in bold).

Model	S2S error (in mm)						
	Top	T-shirt	Trousers	Jumpsuit	Skirt	Dress	All
CLOTH3D++ baseline [30]	23.9	43.3	40.6	22.1	32.8	29.3	31.3
SMPLicit [12]	41.4	36.2	31.8	-	68.1	-	42.9
Ours	**18.1**	**21.6**	**15.3**	**18.3**	**25.6**	**21.1**	**19.1**

In Fig. 5, we show a qualitative comparison between our solution and SMPLicit. We observe that both models produce realistic results and capture well the shape of the garments. SMPLicit generates smoother garment surfaces than our model but is not able to capture all garment dynamic details. Moreover, as shown in the first row, SMPLicit cannot reconstruct properly skirts that have large displacements over the body, since it was trained with skirts very tight to the bodies.

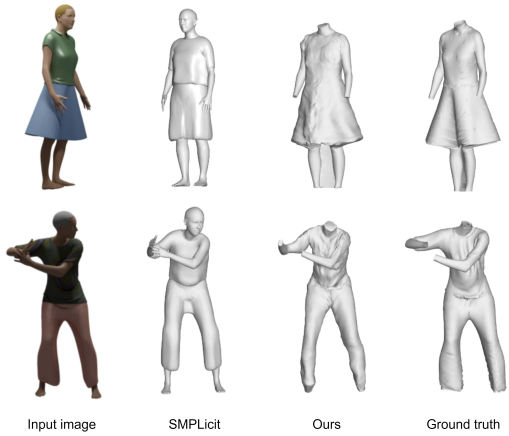


Fig. 5: Qualitative comparison against SMPLicit [12].

V. CONCLUSION

In this paper, we have proposed a system that learns garment dynamics and reconstructs 3D garments from a single RGB image, by using UV maps to represent 3D data. Our model achieves state-of-the-art results on the CLOTH3D++ dataset, generating good quality and realistic reconstructions and being able to deal with lighting, occlusions, viewpoint, dynamism, etc. It has the capacity to learn the dynamics of garments and infer their shape regardless of their topology and the human's action, pose, gender, race and body shape. Moreover, with just a single network we are able to reconstruct a wide variety of different types of garments including loose-fitting ones.

As future work, we could take advantage of the temporal data to improve the current performance by using an attention-based mechanism. We could also try to control the generation of our model with the conditioning it allows, and test it in the task of garment shape transition and editing.

VI. ACKNOWLEDGMENTS

This work has been partially supported by the Spanish project PID2019-105093GB-I00 (MINECO/FEDER, UE) and CERCA Programme/Generalitat de Catalunya.) This work is partially supported by ICREA under the ICREA Academia programme and Amazon Research Awards.

REFERENCES

[1] T. Alldieck, M. Magnor, B. L. Bhatnagar, C. Theobalt, and G. Pons-Moll. Learning to reconstruct people in clothing from a single rgb camera, 2019.

[2] T. Alldieck, G. Pons-Moll, C. Theobalt, and M. Magnor. Tex2shape: Detailed full human body geometry from a single image, 2019.

[3] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. 06 2007.

[4] M. Bertalmio, A. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. volume 1, pages I–355, 02 2001.

[5] H. Bertiche, M. Madadi, and S. Escalera. Cloth3d: Clothed 3d humans, 2020.

[6] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll. Multi-garment net: Learning to dress 3d people from images, 2019.

[7] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image, 2016.

[8] D. Bradley, T. Popa, A. Sheffer, W. Heidrich, and T. Boubekeur. Markerless garment capture. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 27(3):99, 2008.

[9] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, Dec 2016.

[10] P. Chen, Y. Chen, D. Yang, F. Wu, Q. Li, Q. Xia, and Y. Tan. I2uv-handnet: Image-to-uv prediction network for accurate and high-fidelity 3d hand mesh modeling, 2021.

[11] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction, 2016.

[12] E. Corona, A. Pumarola, G. Alenyà, G. Pons-Moll, and F. Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people, 2021.

[13] H. Fan, H. Su, and L. Guibas. A point set generation network for 3d object reconstruction from a single image, 2016.

[14] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou. Joint 3d face reconstruction and dense alignment with position map regression network, 2018.

[15] R. Girshick. Fast r-cnn, 2015.

[16] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[18] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation, 2018.

[19] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. Meshcnn. *ACM Transactions on Graphics*, 38(4):1–12, Jul 2019.

[20] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[22] M.-H. Jeong, D.-H. Han, and H.-S. Ko. Garment capture from a photograph. *Comput. Animat. Virtual Worlds*, 26(3–4):291–300, May 2015.

[23] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose, 2018.

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[25] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop, 2019.

[26] N. Kolotouros, G. Pavlakos, and K. Daniilidis. Convolutional mesh regression for single-image human shape reconstruction, 2019.

[27] A. Laurentini. The visual hull concept for silhouette-based image understanding. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16:150–162, 03 1994.

[28] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on $\mathcal{X}$-transformed points, 2018.

[29] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[30] M. Madadi, H. Bertiche, W. Bouzouita, I. Guyon, and S. Escalera. Learning cloth dynamics: 3d + texture garment reconstruction benchmark. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track, PMLR*, volume 133, pages 57–76, 2021.

[31] M. Madadi, H. Bertiche, and S. Escalera. Smplr: Deep smpl reverse for 3d human pose and shape recovery, 2019.

[32] D. Maduskar and N. Dube. Navier–stokes-based image inpainting for restoration of missing data due to clouds. In M. K. Sharma, V. S. Dhaka, T. Perumal, N. Dey, and J. M. R. S. Tavares, editors, *Innovations in Computational Intelligence and Computer Vision*, pages 497–505, Singapore, 2021. Springer Singapore.

[33] P. Mandikal and R. V. Babu. Dense 3d point cloud reconstruction using a deep pyramid network, 2019.

[34] M. Mirza and S. Osindero. Conditional generative adversarial nets, 2014.

[35] R. Natsume, S. Saito, Z. Huang, W. Chen, C. Ma, H. Li, and S. Morishima. Siclope: Silhouette-based clothed people, 2019.

[36] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Laplacian mesh optimization. GRAPHITE '06, page 381–389, New York, NY, USA, 2006. Association for Computing Machinery.

[37] M. Omran, C. Lassner, G. Pons-Moll, P. V. Gehler, and B. Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation, 2018.

[38] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image, 2019.

[39] D. Pavllo, G. Spinks, T. Hofmann, M.-F. Moens, and A. Lucchi. Convolutional generation of textured 3d meshes, 2020.

[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017.

[41] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017.

[42] K. Regmi and A. Borji. Cross-view image synthesis using conditional gans, 2018.

[43] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions, 2017.

[44] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization, 2019.

[45] S. Saito, T. Simon, J. Saragih, and H. Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization, 2020.

[46] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[47] D. Smith, M. Loper, X. Hu, P. Mavroidis, and J. Romero. Facsimile: Fast and accurate scans from an image in less than a second, 2019.

[48] H. Tang, D. Xu, Y. Yan, P. H. S. Torr, and N. Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation, 2020.

[49] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs, 2017.

[50] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid. Bodynet: Volumetric inference of 3d human body shapes, 2018.

[51] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images, 2018.

[52] C. Wen, Y. Zhang, Z. Li, and Y. Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation, 2019.

[53] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, 2017.

[54] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019.

[55] X. Xu, H. Chen, F. Moreno-Noguer, L. A. Jeni, and F. D. la Torre. 3d human shape and pose from a single low-resolution image with self-supervised learning, 2020.

[56] B. Zhou, X. Chen, Q. Fu, K. Guo, and P. Tan. Garment modeling from a single image. *Comput. Graph. Forum*, 32:85–91, 2013.