Figure 1: Using just small a collection of monocular videos captured with a mobile phone (left), we propose a novel approach to reconstruct the 3D garment layer with fine-scale details (center), and use them to learn a deformable model of the captured apparel (right). Our learned model enables to animate the captured garment, exhibiting material-specific details, as a function of the underlying body pose.

Abstract
Clothing plays a fundamental role in digital humans. Current approaches to animate 3D garments are mostly based on realistic physics simulation; however, they typically suffer from two main issues: high computational run-time cost, which hinders their deployment; and simulation-to-real gap, which impedes the synthesis of specific real-world cloth samples. To circumvent both issues we propose PERGAMO, a data-driven approach to learn a deformable model for 3D garments from monocular images.

To this end, we first introduce a novel method to reconstruct the 3D geometry of garments from a single image, and use it to build a dataset of clothing from monocular videos. We use these 3D reconstructions to train a regression model that accurately predicts how the garment deforms as a function of the underlying body pose. We show that our method is capable of producing garment animations that match the real-world behavior, and generalizes to unseen body motions extracted from motion capture dataset.

CCS Concepts
• Computing methodologies → Computer graphics; Neural networks;

1. Introduction

Synthesizing realistic 3D digital humans is a key topic in Computer Animation due to the large number of applications in industries such as video games, films, telecommunications, and more. A fundamental challenge in digital humans is the modeling of dynamic clothing and garments, because it requires solutions capable of representing large diversity of designs, complex materials, and highly deformable surfaces.

The most well-established approach to model clothing is physics-based simulation, which is today a mature field capable of generating highly-detailed results [NSO12, Stu18, CLMMO14].

However, simulation methods suffer from two main difficulties that hinder their deployment in plug-and-play use cases: first, the significant run-time computational cost, which hinders achieving real-time frame rates. Current solutions for fast simulation need to simplify dynamics [BMO“14] or to adopt custom GPU solutions [TWL“18, LTT“20]; and second, the simulation-to-real gap, which prevents to synthesize simulated cloth that behaves exactly as a desired real-world target. Current methods attempt to fit simulation parameters to real-world observations, but require controlled setups and are limited to small patches of fabrics [MBT“12].

Alternatives to physics-based cloth methods have been proposed to independently address these shortcomings. On one hand, impressive advances in data-driven methods based on neural networks [SOC19, PLPM20, CMM“20] can highly-efficiently infer garment deformations for parametric avatars, generating much faster animations compared to physics simulations. However, training data
is usually obtained with simulation, hence the simulation-to-real gap remains. On the other hand, accurate 3D reconstruction methods [MPHB17, XPB+21] can recover highly-detailed real-world garments, which can be potentially be used to train data-driven approaches. However, the need for multi-camera professional setups prevents to democratize such strategies. All in all, modeling fast and real-world realistic garment deformations remains an open challenge with existing methods.

To overcome all these limitations, we introduce PERGAMO, an approach to learn a deformation model for 3D garments from a single monocular video. PERGAMO is based on two key features: it is learned from casual real-world images, hence there is no simulation-to-real gap or need for multi-camera setups; and it is highly-efficient to evaluate, since at inference time it uses a shallow neural network that directly outputs garment deformations. All in all, our main contribution is a 3D clothing reconstruction pipeline that is able to recover the explicit layer of a garment from just monocular RGB input. These reconstructions enable us to train a data-driven model to infer how a specific garment deforms.

To formulate PERGAMO, we use a novel two-stage approach where we first build a dataset by reconstructing the 3D geometry of deformed garments (Section 4), and then learn a nonlinear regressor from the reconstructed meshes. More specifically, we initially extract human-related features such as body segmentation, body pose, and body normals, from the input images (Section 4.1), which we leverage to deform a mesh template to reconstruct fine-scale detailed clothing using a differentiable rendering optimization (Sections 4.2 and 4.3). Then, we use the reconstructed garments as ground truth data to train a 3D garment deformation regressor (Section 5). We show that the learned model outputs pose-dependent garment surface details, such as folds and wrinkles, that closely match the real-word behaviour of the garment.

2. Related Work

Our work is related to the areas of 3D reconstruction and animation of garments and humans. In this section we discuss the existing literature in these areas.

Multi-Camera 3D Garment Reconstruction. Many works on 3D garment reconstruction require multi-camera setups to capture the scene from different viewpoints. Initial approaches use studio setups, controlled environment, and color-coded fabrics [SSK*05, WCF07] to resolve depth and appearance ambiguities. Follow-up methods lifted the need for color patterns by, for example, using photometric stereo and controlled color lighting [HVB’07], or using multi-view stereo [BPS’08]. This enabled the 3D reconstruction of markerless clothing in controlled multi-camera settings, but output meshes lack high-frequency details such as folds and wrinkles. To mitigate this, some methods use postprocessing steps to add wrinkle detail. For example, Popa et al. [PZB*09] add spatio-temporal coherent wrinkle details to the 3D reconstructed meshes by detecting edges in the original images. Wu et al. [WVL*11] leverage shading information in general unconstrained lighting conditions to add fine-scale dynamic wrinkles. Alternatively, Lähner et al. [LCT18] use an image-to-image translation network based on a Generative Adversarial Network (GAN) to add high-frequency detail to a normal map.

Assuming a detailed point cloud, typically obtained from sophisticated multi-camera setup, it greatly facilitates the 3D garment reconstruction task. For example, ClothCap [MPHB17] reconstructs fine 3D garment geometry as well as the underlying body. Similarly, Xiang et al. [XPB+21] leverage an expensive 140 camera setup to reconstruct dressed humans, including an explicit clothing layer. Bhatnagar et al. [BTTPM19] demonstrate that, given a large dataset of 3D scans, it is also possible to learn to infer 3D garments geometry from images. Bang et al. [BKL21] estimate accurate garment 2D patterns from full body scans, which enables to re-animate the recovered clothing using simulation. We follow a similar goal, but propose a method that only requires a single uncalibrated camera.

Monocular 3D Garment Reconstruction. To democratize the 3D reconstruction of garments, some methods have focused on relaxing the capture requirements and enable the reconstruction from single RGB image [ZCF*13, CZL*15, DDÖ*17, JZH*20, YPA*18, SWY*22]. Zhou et al. [ZCF*13] combine human pose estimation, garment outline prediction, and shape-from-shading cues to reconstruct garments from monocular images. Daněk et al. [DÖ*17] propose a learning-based solution to directly estimate garment vertex displacements from single images. Similarly, Jiang et al. [JZH*20] also learn to infer parametric garments ready to be worn by a SMPL [LMR*15] human model. Other methods require complex physics simulation steps [YPA*18, SWY*22] to drive the surface reconstruction such that it matches the input image. We also reconstruct a garment layer from single image input, but we achieve significantly higher wrinkle detail and do not require physics simulation. Key to our success is the combination of state-of-the-art normal map prediction with a differentiable rendering scheme to optimize vertex positions.

To reduce the inherent depth ambiguity in monocular RGB images, some methods resort to depth or RGB-D input to reconstruct garments. For example, Chen et al. [CZL*15] combine garment parsing and contextual priors to reconstruct garments in rest pose from Kinect-based input. Yu et al. [YZZ*19] combine RGB-D input with physics-based simulation to reconstruct 3D garments. Despite the promising results, requiring depth input hinders the use of these approaches in domestic environments.

Multi-Camera 3D Human Reconstruction. Our work is also related to the methods that, instead of reconstructing an explicit mesh to represent the garment, aim at reconstructing a fully dressed body using a single mesh [SH07, VBMP08, DAST*08, SGDA*10, RCR*16]. This line of research, often referred to performance capture, usually employs multi-camera setups, and enables the replay of captured performances from any viewpoint. Methods within this category can be split into model-based [DAST*08, VBMP08, SGDA*10], which deforms a template to fit it into the images, and model-free [SH07, CCS*15, WWV*16], which have no prior knowledge of the shape and estimate it using the visual hull given by multi-view silhouettes. Despite the high realism of the reconstructions achieved by these methods, they suffer from two main limitations: first, clothing and body are reconstructed in a single
Monocular 3D Human Reconstruction. Many methods have been proposed to directly reconstruct dressed 3D humans as a single 3D mesh (i.e., no explicit modeling of a garment layer) from monocular RGB [ZYW∗19, HXZ∗20, XPWH20, MYR∗20, AMB∗19] or RGB-D input [BNT21, YGX∗17]. Most of these works build on top of the impressive advances in deep learning for human pose estimation [MSS∗17] and parametric model fitting [KB18, PZD18] in single images.

A common strategy in RGB methods is to use a 3D template of the subject, and deformed it to match visual cues (e.g., silhouette, 2D joints, etc.) extracted from the input image [XCD∗18]. Custom GPU-based optimizers have enabled such methods to run at interactive frame rates [HXZ∗19] and, more recently, data-driven methods have been trained to directly infer the geometry of dressed humans [HXZ∗20]. Despite the progress made by these methods, many challenges remain: requiring a subject template is not ideal, output meshes tend to exhibit baked-in garment details that remain static across the poses [GCSH21], and most importantly, they do not explicitly separate garment from body.

Closest to ours is MonoClothCap [XPWH20], an RGB method that does not require a subject-specific template. Instead, a generic parametric human model [LMR∗15] is deformed using a differentiable rendering scheme to fit into per-frame estimated normals of an actor. We follow a similar path, but we are able to reconstruct an explicit layer of the garment, which subsequently allows us to train a regressor to predict how such garment deforms given arbitrary body motions.

As an alternative representation for 3D humans, recent works have explored the use of learning-based implicit functions [PFS∗19]. This representation is continuous, compact, and differentiable, which has enabled to efficiently and accurately reconstruct humans from single RGB images [SHN∗19, SSSJ20, ZYL21] or partial point clouds [BSTP20, PBT∗21]. Despite the impressive advances in this domain, we focus our work on mesh-based representations and explicit garments, which directly enables animation of clothing in well-established animation pipelines.

3D Garment Animation. Many works exist that address the problem of creating 3D animatable garments and dressed avatars. The key underlying challenge is to learn a model that is able to infer how the surface of the garment deforms as a function of the body pose and/or shape. Earlier works assume multi-camera footage [XLS∗11, CVCH14], and do not model clothing layer independently. Assuming accurate 3D reconstructed meshes of dressed humans, some methods have tackled the modeling of garment deformations as a function of the underlying body [YFHWW18, NH14, PMPH17]. Our work tackles a similar problem, but learns to deform a garment layer from monocular RGB.

With the rise of data-driven methods to learn highly nonlinear problems, recent works have attempted to model the deformation of garments leveraging large datasets of simulated data [CMM∗20, JZGF20]. Such ground-truth dataset are usually generated offline using computationally expensive physics-based simulations [NSO12]. Closest to ours, Santesteban et al. [SOC19] use a recurrent neural network to learn shape-and-pose dynamic deformations of a single garment. Similarly, TailorNet [PLPM20] predicts garment deformations also as a function of clothing style, and decomposes deformation into a high frequency and a low frequency component to improve the detail. Vidaurre et al. [VSGC20] show better generalization capabilities to garment design by leveraging a fully convolutional architecture capable of inferring deformations for any mesh topology. Despite great results, these methods require loss supervision at the vertex level. Alternatively, very recent methods demonstrate that it is possible to learn deformations directly from physics-based losses [BME21, SOC22] or using distance fields [CPA∗21].

Animation of 3D garments has also been tackled using clothed humans represented as a single mesh. CAPE [MYR∗20] uses detailed 4D scans of dressed avatars to learn a graph convolutional neural network to deform a full body mesh template. More recent works use implicit representations [SYMB21, WM∗21] and are capable of encoding finer details. Nonetheless, using a single mesh to encode body and garment is not ideal for animation. Instead, we focus on learning an model for garments using an explicit mesh 3D layer. As we demonstrate, this enables to dress animated characters using existing motion capture datasets such as AMASS [MGT∗19].

3. Overview

Our goal is to learn a deformable model for 3D garments that faithfully reproduces the behavior of a real garment-subject pair. In other words, we want to model how a specific garment deforms when worn by a particular subject. This effectively enables a personalized animation of garments for many tasks including: virtual try-on, telepresence, VR, videogames, and more.

More formally, we want to learn a model \( R \) that outputs a garment

\[
X = R(\theta),
\]

where \( X \in \mathbb{R}^{N_v \times 3} \) encodes the vertices of the deformed garment in rest pose (i.e., it contains pose-dependent wrinkles and folds), and \( \theta \) is the target pose. In practice, the garment \( X \) is worn by a parametric human body such as a SMPL [LMR∗15] that represents the body surface of the target subject. To rig the deformed garment \( X \), we borrow the skinning functionalities (e.g., skeleton, rigging weights) from the underlying body model.

Given our goal for personalized garments (i.e., specific real-world behavior), the ground-truth data to learn \( R() \) cannot be obtained with physics-based cloth simulation tools as done in state-of-the-art methods [SOC19, PLPM20, VSGC20, BME21], for two reasons: first, due the unknown mechanical properties of the target garment, which are hard to obtain even in complex laboratory setups [MBT∗12]; and, second, due to the unknown soft-tissue model of the underlying body, which also effects how garment behave [RCP20]. These limitations preclude generating synthetic data that match specific real-world behavior.

Therefore, instead of learning to deform garments using 3D simulations, we propose a image-driven approach that learns to infer
the true behaviour of a garment from videos. To this end, in Section 4 we introduce a method to build a database of 3D meshes by directly reconstructing garments from a set of unconstrained monocular videos. Then, in Section 5, we describe a neural regressor that learns to faithfully deform a 3D garment given a target pose of the subject.

4. 3D Garment Reconstruction

To obtain the dataset that we used to learn our personalized 3D deformation model, we directly reconstruct the surface of the target garment captured from monocular videos. Figure 2 illustrates our reconstruction pipeline.

Since recovering 3D geometry from a single view is an ill-posed problem, we tackle this task in three steps. First, in Section 4.1, we extract a combination of semantic and 3D features from the input image. This information is key to reduce the complexity inherent to our problem. Second, in Section 4.2, we fit a coarse parametric garment model into the input image by leveraging the extracted features. And finally, in Section 4.3, we add fine wrinkle details into the garment using a differentiable rendering scheme.

4.1. Extracting Semantic and 3D Information

Given a monocular video sequence \( I = \{I_t\}_{t=0}^T \), where \( I_t \) is an RGB frame, captured in an uncontrolled setting (i.e., unknown camera parameters, unknown illumination, unknown subject pose), we initially extract a set of human-related image features for each frame \( I_t \). The extracted information will ease our ill-posed 3D reconstruction task that we describe later in the rest of this section.

First, we leverage a state-of-the-art human parsing approach [LXWY20] to assign a per-pixel label to the input image \( I \) (to simplify notation, we drop the subindex \( t \) in the rest of this section, but note that all steps are done per each frame). As output, we obtain a set of binary segmentation masks \( S_t \), one for each label type, that encodes what body part or garment type (e.g., t-shirt, trousers, head, right/left arm, etc.) each pixel contains. For the results shown in this work, we use the binary mask \( S_8 \), which corresponds to the upper body clothing.

Second, we use an image-to-image translation network, based on the work of Saito et al. [SSSJ20], to estimate the surface normals \( N \) for each pixel of the input image \( I \). Notice that estimated normals for pixels in the background can be unreliable or noisy, but since our goal focuses only on garment regions, we do not suffer from these artifacts.

Finally, we estimate the pose \( \theta \in \mathbb{R}^{69} \) of the subject in the input image \( I \) using a state-of-the-art human pose estimation method [CPB20].

4.2. Parametric Garment Fitting

To reconstruct a coarse approximation of the 3D geometry of a deformed garment that appears on an input frame \( I \), we first use a model-based strategy. To this end, we use a parametric garment model

\[
G_{\text{coarse}}(p, \theta) = W(D(p), \theta, W),
\]

where \( W \) is a skinning function (e.g., linear blend skinning) that articulates a parametric garment template \( D(p) \in \mathbb{R}^{25} \times 3 \) based on weights \( W \). Our parametric garment template \( D(p) \) is learned from a public dataset of 3D garments [SOC19] using a variational autoencoder network, where \( p \in \mathbb{R}^{25} \) is the latent space learned with the autoencoder, and \( D() \) is the decoder block. Intuitively, \( p \) encodes a latent representation of T-shirt deformations, and \( D() \) is the mapping from latent variable to vertices position of a template mesh. This parametric template encodes coarse (i.e., not specific to the material of the target garment) deformations of garments in rest pose. Hence, this first fitting step aims at recovering a coarse version of the garment visible in the input frame \( I \).
In order to fit the garment model of Equation 2 to a frame $I$, we leverage the human-related image features $S_g$, $N$, and $\theta$ described in Section 4.1, and formulate the following optimization problem

$$\arg\min_{P} \mathcal{E}_{\text{coarse}} + \mathcal{E}_{\text{sil}} + \mathcal{E}_{\text{temp}} + \mathcal{E}_{\text{reg}}. \quad (3)$$

$\mathcal{E}_{\text{coarse}}$ is the main data term, and enforces the normals of the fitted parametric garment to match the predicted normals $N$. This term is formulated as

$$\mathcal{E}_{\text{coarse}} = \lambda_{\text{coarse}} \| \phi_N(G(p, \theta)) - (N \odot S_g) \|^2 \quad (4)$$

where $\phi_N$ is a differentiable rendering function that outputs camera-space per-pixel normals of the garment $G(p, \theta)$ and $\odot$ is the Hadamard product (i.e., element-wise multiplication).

In practice, the function $\phi_N$ is implemented using the state-of-the-art differentiable rendering library Kaolin [FTSL'22], which enables the computation of gradients of the image error w.r.t. vertices position.

$\mathcal{E}_{\text{sil}}$ is a data-term that enforces the silhouette of the fitted garment to match the predicted mask $S_g$,

$$\mathcal{E}_{\text{sil}} = \lambda_{\text{sil}} \| \phi_S(G(p, \theta)) - S_g \|^2, \quad (5)$$

where $\phi_S$ is a differentiable rendering function that outputs a mask with the silhouette of the garment (with 1s indicating inside, 0s outside and bit of a smooth transition on the edges) in camera space, also implemented using the rendering library Kaolin [FTSL'22]. $\mathcal{E}_{\text{temp}}$ and $\mathcal{E}_{\text{reg}}$ are regularizers formulated as

$$\mathcal{E}_{\text{temp}} = \lambda_{\text{temp}} \| p_t - p_w \|^2 \quad (6)$$

$$\mathcal{E}_{\text{reg}} = \lambda_{\text{reg}} \| p_w \|^2 \quad (7)$$

that enforce temporal stability and avoid unnatural deformations, respectively.

### 4.3. Fine Wrinkle Extraction

The parametric garment model fitted in Section 4.2 is only capable to represent coarse details of the garment. In order to add personalized details (e.g., material-specific wrinkles, pose-and-shape-depending details, etc.), in this final reconstruction step we compute a vector $V \in \mathbb{R}^{N \times 3}$ of per-vertex 3D displacements to add fine details. To this end, we formulate our final garment model as

$$G_{\text{fine}}(p, \theta, V) = W(D(p) + V, \theta, W), \quad (8)$$

and our goal in this last reconstruction step is to find the vector of 3D displacements $V$ for each frame we want to reconstruct. Notice that the rest of parameters are already known.

Similar to the parametric garment fitting step from Section 4.2, we find the 3D displacements vector $V$ solving an optimization at each frame

$$\arg\min_{V} \mathcal{E}_{\text{fine}} + \mathcal{E}_{\text{edge}} + \mathcal{E}_{\text{temp}} + \mathcal{E}_{\text{reg}} \quad (9)$$

s.t. $\forall v_i \in V : \eta_{\text{min}} < v_i < \eta_{\text{max}}$,

where $\eta_{\text{min}}$ and $\eta_{\text{max}}$ are thresholds for minimum and maximum allowed displacements, respectively. $\mathcal{E}_{\text{fine}}$ is the main data term, and it resemble to the term $\mathcal{E}_{\text{coarse}}$ from Equation 4, but with two important differences. Specifically,

$$\mathcal{E}_{\text{fine}} = \lambda_{\text{fine}} \cdot \| (\nabla \phi_N(G_{\text{fine}}(p, \theta, V)) - \nabla (N \odot S_g)) \odot \phi_S(G(p, \theta)) \|^2 \quad (10)$$

where $\nabla$ is the image gradient operator, $\odot$ is the Hadamard product (i.e., element-wise multiplication), and $\phi_N(G(p, \theta))$ a function that renders the silhouette of the parametric garment fitted in the previous section. In practice, the $\mathcal{E}_{\text{fine}}$ term allows the vertices of the garment to move freely to match the image normal $N$, up to fine details to reproduce wrinkles. The gradient image operator $\nabla$ ensures attention on the parts of the image normal where normals change (e.g., the wrinkles). Finally, the element-wise multiplication constrains this optimization to focus only on the pixels of the image where the garment $G(p, \theta)$ appears.

Equation 10 allows free vertex movement, however, many of the garment vertices are not visible from the input image due to occlusions and self-occlusions. Allowing occluded vertices to move can potentially lead to undesired reconstruction artefacts. To resolve this ill-posed problem, we rely on a set of regularizers to constrain the optimization. Namely, we use the following terms

$$\mathcal{E}_{\text{edge}} = \lambda_{\text{edge}} \| E_{\text{est}} - E_r \|^2 \quad (11)$$

$$\mathcal{E}_{\text{temp}} = \lambda_{\text{temp}} \| V_{t-1} - V_t \|^2 \quad (12)$$

$$\mathcal{E}_{\text{reg}} = \lambda_{\text{reg}} \| V \|^2, \quad (13)$$

where $E_{\text{est}}$ and $E_r$ are the edge lengths of the garment in rest pose and the optimized garment, and $V$ the optimized garment vertices. These three terms temporally and spatially regularize the garment deformation, such that the reconstructed mesh does not exhibit unnatural deformations.

### 5. 3D Garment Regressor

The method introduced in Section 4 allows us to create a dataset of 3D reconstructed meshes $M$, each of them with an associated pose parameter $\theta$. In this section we describe how to use this data to learn the model defined in Equation 1, capable of inferring pose-dependent deformations. Notice that the model proposed here is independent of the reconstruction method, and it could potentially be used on other similar 3D garment datasets.

We follow the standard approach in data-driven garments [SOC19], and convert our dataset of reconstructed meshes into a dataset of per-vertex displacements with respect to a template mesh in T-pose. Specifically, for each mesh $M$ of our dataset we compute ground truth displacements

$$\Delta_{\text{GT}} = W^{-1}(M, \theta, W) - \bar{T}, \quad (14)$$

where $W^{-1}$ is the inverse skinning transformation, $\theta$ the pose parameter, and $\bar{T}$ is the average unposed garment of the dataset. The final deformed garment is then defined as $X = \bar{T} + \Delta_{\text{GT}}$.

To formulate the model defined in Equation 1, we use a regressor that is trained to predict per-vertex garment offsets $\Delta$ as a function of the target body pose $\theta$. In practice, we implement it using a simple MLP network with 3 hidden layers, supervised with an L1 loss.
on vertex offsets and vertex normals:

\[
\mathcal{L} = L_1(\Delta, \Delta_{GT}) + L_1(N(\bar{T} + \Delta), N(\bar{T} + \Delta_{GT}))
\]

(15)

where \(N()\) is a function that computes the per-vertex normals of the garment meshes. Check Section 6.3 for further implementation details.

In contrast to existing data-driven methods that are trained on hundreds of simulated meshes [PLPM20, SOC19, BME20], we only have a reduced set of reconstructed sequences which significantly complicates the learning. To ease the generalization capabilities of our regressor despite relatively small training set, we encode the pose vector \(\theta\) in a compact subspace that better captures key pose features. To this end, we leverage the multi-modal autoencoder introduced by SoftSMPL [SGOC20], which encodes poses in \(\mathbb{R}^{10}\). Hence, our final regressor is \(R() : \mathbb{R}^{10} \rightarrow \mathbb{R}^{N_G \times 3}\), where \(N_G\) is 4,424 for the garment showcased in our results.

Finally, notice that due to residual errors at inference time, a few garment vertices might collide with the underlaying body mesh. Similar to existing works in data-driven garments [SOC19, PLPM20], we push problematic vertices the normal direction of the closest body point.

6. Results and Evaluation

6.1. Garment 3D Reconstruction

Qualitative Evaluation

Figure 3 and the supplementary video showcase a large variety of results of our 3D garment reconstruction pipeline introduced in...
Section 4. To visually stress the quality of our results, we show side-by-side the corresponding input frame for each reconstruction.

We have processed a dataset consisting in 3 different garments—grey, yellow, and brown t-shirts—, and 12 motion sequences for each garment. Importantly, each t-shirt is made of a different fabric, hence exhibiting different folds and wrinkles. To highlight this, each row in Figure 3 depicts representative frames of a different garment, and each column similar poses. It can be observed that the yellow t-shirt—a thin 100% polyester sports t-shirt—produces fine wrinkles; the grey t-shirt—100% cotton t-shirt—produces mid-scale wrinkles; and the brown t-shirt—made with thick fabric, similar to a sweater—produces coarser wrinkles.

Additionally, in Figure 4 and in the supplementary video, we show qualitative results of our method in sequences of the Panoptic CMU dataset [XJS19, JSS18]. Notice that MonoClothCap is a monocular reconstruction approach that outputs clothed avatars encoded in a single mesh, which is not ideal to model garments. Results demonstrate that our approach is capable of recovering finer wrinkle detail, while reconstructing an explicit garment layer.

![Figure 4: Qualitative comparison to MonoClothCap [XPWH20] in CMU Panoptic Dataset [XJS19, JSS18]](image)

Similarly, in Figure 5 we qualitatively compare our reconstruction results to both MonoClothCap [XPWH20] and MonoPerfCap [XCZ18] in an outdoor sequence. The visual quality of our reconstruction outperform these works, while we are able to also recover an explicit layer of the garment.

![Figure 5: Qualitative comparison to MonoClothCap [XPWH20] and MonoPerfCap in Oleksander_outdoor sequence from [HXZ19]](image)

Quantitative Evaluation

Recovering 3D geometry from a single view is an ill-posed problem. To tackle it, we use multiple regularization terms in our reconstruction pipeline. In Figure 6 and 8 we depict the effects of these terms, which we discuss below.

We first analyze the effect of the terms $\mathcal{E}_{temp}$ and $\mathcal{E}_{reg}$ of our parametric garment fitting step (Section 4.2). To this end, we use sequences of the BUFF dataset [ZPBPM17] which provides sequences of 3D human textured meshes. To use such mesh data in our image-based pipeline, we render RGB and ground-truth surface normal in a $512 \times 512$ pixels image. We then process the resulting RGB frames using our method and a variety of values for $\lambda_{temp}$ and $\lambda_{reg}$. In Figure 6 we show representative frames of the sequence shortlong_hips_95 reconstructed using different regularization choices (rows one to four). The first row shows how the mesh may degenerate when no regularization is enforced. The second row shows how a slight regularization of the VAE latent space cannot completely prevent the mesh degeneration, while inconsistencies between consecutive frames may appear causing flickering artifacts (see differences between frame 89 and 90). Temporal regularization (third row) can prevent flickering artifacts at the expense of a worse fit to the ground-truth silhouette. A careful combination of both regularization strategies may prevent degeneration and flickering while not greatly affecting the silhouette fitting.

We also analyze the effect of regularization mechanisms implemented at the fine wrinkle extraction step (Section 4.3). Row 5 at Figure 6 shows how the optimization process can explode when no regularization on the free-vertex movements is enforced. Finally,
6.2. Garment 3D Regressor

To validate that the garment deformation regressor introduced in Section 5 generalizes to human poses not present in our dataset, we test the regressor using publicly available motion capture data. Our goal is to demonstrate that our model can produce realistic clothing animations for arbitrary motion input.

To this end, we use motion sequences from the publicly available AMASS [MGT+19] and BUFF [ZPBPM17] datasets, and feed them into a regressor trained with the grey t-shirt dataset. Figure 7 and the supplementary video show realistic animations generated with our approach.

6.3. Implementation Details

In this section we provide additional implementation and performance details of our method. We have implemented PERGAMO in a desktop machine with an Intel Core i7-5820K CPU, a Nvidia RTX 3080 Ti GPU and 16GB of DDR4 RAM.

Parametric Garment Fitting (Section 4.2). In our experiments we set $\lambda_{\text{reg}} = 1$ and start with $\lambda_{\text{coarse}} = 0$ and increase it to 1 after half of the optimization iterations. In particular, we run the optimization loop for a total of 200 iterations for the first frame of a sequence using gradient descent. Afterwards, we always initialize $p_t$ using $p_{t-1}$ and, thanks to this, we can reduce the number of optimization iterations to 20. We use regularization weights $\lambda_{\text{empy}} = 1$ and $\lambda_{\text{ref}} = 0.01$, and explore the effect of these in the ablation study conducted in Section 6.1.

Fine Wrinkle Extraction (Section 4.3). We fix $\lambda_{\text{inew}} = 6.5 \cdot 10^{-5}$ and use regularization weights $\lambda_{\text{empy}} = 10^6$, $\lambda_{\text{edges}} = 10^4$ and $\lambda_{\text{ref}} = 0.1$. We explore the effect of these in the ablation study conducted in Section 6.1. In our experiments we run the optimization loop for a total of 200 iterations for the first frame of a sequence using gradient descent. Afterwards, we always initialize $V_t$ using $V_{t-1}$ which allows us to reduce the number of optimization iterations to 20.

The values for the different loss parameters $\lambda$ have been set empirically using trade-off between the faithfulness and stability of the reconstructions. Selected values have been used for all reconstructions shown in this paper, including the sequences for the 3 different types of shirts, the Panoptic CMU Dataset [JSS18], and MonoPerfCap [XCGZ18] sequences used for providing comparisons.

Our full reconstruction pipeline for one input frame takes around 2,615 ± 0.039 seconds.

Garment Regressor (Section 5). We use a vanilla MLP ($R() : \mathbb{R}^{10} \rightarrow \mathbb{R}^{N_G \times 3}$) with 3 hidden layers ($HL_1() : \mathbb{R}^{10} \rightarrow \mathbb{R}^{N_0}$, $HL_2() : \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_0}$ and $HL_3() : \mathbb{R}^{N_0} \rightarrow \mathbb{R}^{N_0 \times 3}$, with $N_G = 4,424$ for the garment in our results). Each layer is fully-connected with the previous layer, and the first and second layers are followed by a LeakyRELU activation (slope = 0.1) and a Dropout (10%) layer. We use an Adam optimizer for 100 epochs with batch size of 64, and decreasing learning rate from 5e-3 to 1e-5. Finally, we have noticed that normalizing the inputs and outputs of our regressor using the training set statistics can ease the training process.

After training, predicting the deformations for a single pose takes...
7. Conclusions

We have introduced PERGAMO, a novel approach to learn a deformable model for 3D garments directly from monocular videos. To the best of our knowledge, PERGAMO is the first method to reconstruct an explicit 3D garment layer from single view, and use the reconstructed geometry to learn a deformable model. Since PERGAMO uses training data that comes directly from real-world images, it circumvents the simulation-to-real gap issue that existing data-driven methods suffer. Comparisons to existing methods demonstrate that our approach is capable of recovering finer wrinkle detail, and it generalizes well to unseen motions at train time.

Despite the step forward that PERGAMO does in the field of 3D garment modeling and animation, it still suffers from a few limitations. First, garment self-collisions are not explicitly modeled, therefore, despite that we regularize 3D deformations to avoid geometry artifacts, residual collisions can occur in highly deformed areas such as armpits. Second, our approach is only capable of reconstruction garments that are close to the coarse deformable mesh used in our reconstructing step. It remains open to future research how to generalize the reconstruction pipeline to a larger variety of clothing. Additionally, at the moment we do not model body shape variations, which prevents PERGAMO to generalize to unseen subjects. Third, our method is dependent on the quality of the information used as input. Noisy estimated normals, pose or segmentation can impact the quality of the reconstructed garments. And fourth, the regressor may produce unsatisfactory results when dealing with extreme poses as can be seen in Figure 9.

Acknowledgments

This work has been partially funded by: the Comunidad de Madrid in the framework of the Multiannual Agreement with the Universidad Rey Juan Carlos in line of Action 1, "Encouragement of research for young PhD", project CaptHuRe (M2736); the Universidad Rey Juan Carlos through the Distinguished Researcher position INVESDIST-04 under the call from 17/12/2020; and by a Leonardo Fellowship from the Fundación BBVA.
References


© 2022 The Author(s)
Creating New Human Performances from a Multi-View Video Database. 
doi:10.1145/2010324.1964927. 3

[XPB+21] XIANG D., PRADA F., BAGAUTDINOV T., XU W., DONG Y., 
WEN H., HODGINS J., WU C.: Modeling Clothing as a Separate Layer 
3480545. 2

[XPWH20] XIANG D., PRADA F., WU C., HODGINS J.: Monocloth-
cap: Towards temporally coherent clothing capture from monocular rgb 
332. doi:10.1109/3DV50981.2020.00042. 3, 7

[YFHWW18] YANG J., FRANCO J.-S., HETROY-WHEELER F., 
WUHRER S.: Analyzing Clothing Layer Deformation Statistics of 3D 
Human Motions. In Proc. of European Conference on Computer Vision 

[YG*17] YU T., GUO K., XU F., DONG Y., SU Z., ZHAO J., LI J., 
DAI Q., LIU Y.: BodyFusion: Real-time capture of human motion and 
surface geometry using a single depth camera. In Proceedings of the 
IEEE International Conference on Computer Vision (2017), pp. 910– 
919. 3

[YPA+18] YANG S., PAN Z., AMERT T., WANG K., YU L., BERG T., 
LIN M. C.: Physics-Inspired Garment Recovery from a Single-View 
3026479. 2

[ZC+13] ZHOU B., CHEN X., FU Q., GUO K., TAN P.: Garment mod-
85–91. doi:10.1111/cgf.12215. 2

[ZPBPM17] ZHANG C., PUJADES S., BLACK M. J., PONS-MOLL G.: 
Detailed, accurate, human shape estimation from clothed 3d scan se-
quences. In Proc. of Computer Vision and Pattern Recognition (CVPR) 
(2017), pp. 4191–4200. 7, 8, 9

Model-Conditioned Implicit Representation for Image-based Human 
Reconstruction. IEEE Transactions on Pattern Analysis and Machine 
Intelligence (2021). doi:10.1109/TPAMI.2021.3050505. 3

[ZYW+19] ZHENG Z., YU T., WEI Y., DAI Q., LIU Y.: Deephuman: 3d 
human reconstruction from a single image. In Proc. of IEEE Interna-
tional Conference on Computer Vision (ICCV) (2019). 3