Deep Self Supervised Learning for 3D Surface Parameterization and 3D Garment Retargeting

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering by Research

by

Shanthika Naik 2020701013 shanthika.naik@research.iiit.ac.in



International Institute of Information Technology Hyderabad - 500032, INDIA September, 2023

Copyright © Shanthika Naik, 2023 All Rights Reserved

International Institute of Information Technology Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Deep Self Supervised Learning for 3D Surface Parameterization and 3D Garment Retargeting" by Shanthika Naik, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Dr. Avinash Sharma

Co-Adviser: Dr. KS Rajan

To endless possibilities..

Acknowledgments

I would like to express my heartfelt gratitude and appreciation to my advisor, family, colleagues, and friends for supporting me in successfully completing my thesis. It is with great pleasure and humility that I extend my acknowledgments to the following individuals:

First and foremost, I would like to express sincere gratitude to my guide, Dr. Avinash Sharma. He has played a massive role in helping me shape my life and career, which I didn't imagine myself capable of prior to joining here. I will forever be grateful for the day he accepted me as a student. I'm thankful for his immense patience, for looking past my silly mistakes and guiding me toward improving them. He has provided me with the opportunity to explore different problems and interests before finalizing my thesis statement. His teachings have helped me improve not only in academia but also grow as a person. I am extremely grateful for our discussions and his guidance that has helped me make various decisions in my life. I am looking forward to a great many things that I have yet to learn from him, and hope to continue collaboration in the future.

To my dear family, I am profoundly grateful for your unconditional love, unwavering belief in me, and constant encouragement. I owe everything I am to my dear father, Shankar Naik, who has made innumerable sacrifices for his family. He is my role model, and I hope I can be as cool as him someday. I am extremely thankful for his belief in me and for supporting me in all my decisions. Being from a small town where girls are raised to take care of house chores, I am extremely grateful that he raised me to have a career and be independent. I am also thankful to my mother for her love and support. Lastly, I wish to express my gratitude to my elder brother for his support, for always taking my side, for being strict with me when required, and for encouraging me when I didn't believe in myself.

I would also like to extend my heartfelt appreciation to my peers and colleagues who have supported me throughout this thesis. I am extremely grateful to my senior (in knowledge, not age) Astitva Srivastava, for teaching me various concepts and for having the patience to answer my tiniest of queries. I also wish to thank my dear friend, one of the most smart working people I know, Chandradeep Pokariya, for teaching me how to be disciplined (still have a long way to go), for always being there for me, and for his help which has contributed significantly to the completion of my thesis. I'm also thankful to Kinal Mehta for his care and support and for helping me during the time of submission. I also wish to extend heartfelt gratitude to my colleagues Dhawal Sirikonda, Ishaan Shah, Kunwar Singh, Amogh Tiwari, and Rahul Goel for their help and support. Along with them, I also wish to thank Ritam, Abhinaba, Mounika, Shantanu, Harshit, Siddarth, Soumya, Jeet, and several others who have been great company and have made life here fun and memorable. Last, but not the least, I am extremely grateful to my girl gang, Shreeya, Sarvani, Deepti and Ihita, for always being there, for hearing me out, and for helping me get through my numerous emotional breakdowns.

I would also like to express my deepest appreciation to the academic institution where I conducted this research for providing the necessary resources, infrastructure, and opportunities to pursue my academic aspirations.

To all those whose names I may have inadvertently left out, please know that your contributions have not gone unnoticed. Each and every interaction, big or small, has shaped my research and personal growth in immeasurable ways.

Thank you all from the bottom of my heart.

Abstract

Metaverse has gained massive popularity these days owing to its potential in various industries and research fields such as the health sector, online social interaction, gaming and entertainment, education, eCommerce transactions, etc. Some of the challenges to be addressed in building a Metaverse, including human-computer interactions and scene understanding, rely on AI techniques for solutions. However, most of these methods rely on supervised learning that requires labor-intense annotated data and have limited generalizability. Self-supervised learning methods offer a viable alternative to overcome these limitations as well as leverage large amounts of readily available unlabeled raw data, which is often more abundant and easier to collect. These methods can learn to capture temporal or spatial relationships within data, modeling contextual information, which is valuable for several computer vision tasks. Hence, these methods can be useful in addressing the challenges of building the Metaverse.

Within the Metaverse, users can interact with each other and a digital environment through avatars. Using UV parameterized maps of avatars, textures can be accurately applied, resulting in realistic skin, clothing, and other visual details. They also play a crucial role in the creation of virtual fashion and design within the Metaverse. This can also benefit the E-Commerce industry, particularly fashion and apparel, which have experienced tremendous growth in recent years. The Metaverse can provide immersive virtual environments for virtual try-on where users can explore and interact with products more realistically and engagingly than traditional online shopping.

Given the above advantages, we intend to address the following two challenges in this thesis. First, we explore self-supervised data-driven methods for UV parameterization of general objects. The existing methods for surface parameterization of arbitrary 3D objects face challenges when dealing with closed surfaces and regions of extreme extrinsic curvature. Mapping a surface from 3D to 2D almost always introduces a certain amount of distortion, and the aim is to keep this distortion as low as possible. Finding optimal seams that lead to low distortion is another challenge. We present a novel framework for learning the discretization-agnostic surface parameterization of arbitrary 3D objects with closed and open surfaces. We evaluate our framework on multiple 3D objects from the publicly available dataset and report a comparison with conventional methods.

Secondly, utilizing self-supervised methods, we aim to innovate a solution for a 3D virtual tryon system. The goal is to retarget real, non-parametric garment meshes over a target human body (parametric or non-parametric). This 3D virtual try-on system needs to generalize to arbitrary body shapes and poses, modeling topological differences among various categories of garments, along with realistic deformations arising out of the physical interaction with the underlying body and resolving the penetration/intersection of the garment with the underlying body. These systems need to run in real time with very less delay. We propose a self-supervised method for draping non-parametric, 3D garment meshes by first obtaining the initial alignment between the garment and the human body by establishing correspondences via Isomap Embeddings. Further, this coarse retargeting is refined by training an MLP that preserves the geometry of the garments guided by our novel losses. We propose a wrinkle generation module to obtain realistic details on the draped garments. We also contribute a new dataset of real-world reposed garments with realistic noise and topological deformations.

Finally, we discuss the limitations of our work and lay down the potential solutions that can be explored. We also discuss the future directions that can be pursued based on the findings of our work. We believe this thesis advances the field of virtual try-on systems significantly while providing a learning-based solution for parameterization.

Contents

Ch	apter	P	age			
1	Intro	duction	1			
	1.1	Motivation	2			
	1.2	Problem Statement	6			
		1.2.1 UV parameterisation	6			
		1.2.2 Garment Retargeting	7			
	1.3	Research Landscape	7			
		1.3.1 UV parameterisation	7			
		1.3.2 Garment Retargeting	8			
	1.4	Contribution	8			
	1.5	Thesis Roadmap	9			
2	Back	ground	10			
	2.1	3D Representations	10			
		2.1.1 Point Cloud	11			
		2.1.2 Mesh	11			
	2.2	Parameterisation	12			
	2.3	SMPL: Skinned Multi-Person Linear model	12			
3	Disci	retization-Agnostic Deep Self-Supervised 3D Surface Parameterization	15			
	3.1	Introduction	15			
	3.2 Background					
	3.3	Method	17			
		3.3.1 Patch Extraction Module	18			
		3.3.2 Surface Parameterization Module	20			
		3.3.3 Losses	21			
	3.4	Experiments and Results	22			
		3.4.1 Implementation Details	22			
		3.4.2 EVALUATION METRIC	23			
	3.5	Additional Qualitative Results	23			
	3.6	Results & Evaluation	24			
	3.7	Ablation Study	25			
		3.7.1 Patch Extraction Module	25			
		3.7.2 Surface Parameterization Module	25			
		3.7.3 Effect of Loss Functions:	26			
	3.8	Conclusion	26			

4	Dress Me Up: A Dataset and Method for Self-Supervised 3D Garment Retargeting								
	4.1	Introduction)						
	4.2 Related Work								
	4.3	Method	,						
		4.3.1 Correspondence-Guided Coarse Retargeting)						
		4.3.2 Self-Supervised Refined Retargeting	ł						
		4.3.3 Wrinkle Generation Module	5						
		4.3.3.1 Supervised Wrinkle Generator	5						
		4.3.3.2 Laplacian Detail Transfer	3						
	4.4	Implementation Details)						
		4.4.1 SMPL Registration:)						
		4.4.2 Refined Retargeting Module	L						
		4.4.3 Supervised Wrinkle Generation Module	L						
		4.4.4 Datasets	3						
		4.4.5 Evaluation Metrics	ł						
	4.5	Experimentation and Results	5						
		4.5.0.1 Qualitative and Quantitative Results on CLOTH3D:	5						
		4.5.0.2 Quantitative Results on Our Dataset:	7						
		4.5.0.3 Qualitative Results on Real Scans:	7						
		4.5.1 Comparison)						
		4.5.2 Ablation Studies	L						
		4.5.2.1 Ablation on Self-Supervised Losses	L						
		4.5.2.2 Ablation on Supervised Wrinkle Generation Module	L						
	4.6	Discussion)						
		4.6.1 Description of DressMeUp Dataset)						
		4.6.2 Analysis of Isomap Embeddings)						
		4.6.3 Applications of the Proposed Framework	5						
		4.6.4 Limitations & Future Work	5						
5	Cond	clusion	3						
	5.1	Discussion	3						
	5.2	Impact)						
	5.3	Future Directions)						
Bib	liogr	raphy	L						

List of Figures

Figure	Ι	Page
1.1 1.2	Different applications that use Metaverse	1 3
1.3	Failure case of supervised method for draping parametric 3D garment on SMPL body	
	models	4
1.4	Example of virtual garment try-on system.	5
1.5	The objective is to achieve UV parameterisation and Garment Retargeting	6
2.1	3D representations as point cloud and mesh.	10
2.2	(a) Templete mech with bland weights indicated by color and joints shown in white (b)	12
2.3	(a) remplate mesh with blend weights indicated by color and joints shown in white. (b) With identity-driven blendshape contribution only (c) With the addition of of pose blend	
	shapes in preparation for the split pose: note the expansion of the hips (d) Deformed	
	vertices reposed by dual quaternion skinning for the split pose. Figure adopted from [38]	13
3.1	UV parameterization for open and closed surfaces estimated via our proposed framework.	16
3.2	The outline of proposed framework.	18
3.3	Discretization-agnostic UV parameterization.	19
3.4	Comparison of error plots for QCE and ASE with other methods. First two categories	
	(a) Bird, (b) Pliers are taken from SHREC dataset; (c) Armadillo & (d) Spot	20
3.5	Additional Qualititative Results	21
3.6	Effect of geodesic loss.	23
3.7	Effect of DiffusionNet embeddings.	23
3.8	Patches are used to obtain multiple open surfaces from closed surfaces. As we increase	
	the number of patches, the conformal and angular distortion gets reduced	24
3.9	Ablation study of different losses.	25
3.10	Comparison of global and local embeddings	25
3.11	Few limitations of current method.	26
4.1	3D garment retargeting on real human scans using our approach (left) and our real 3D	
	garment dataset samples(right)	28
4.2	Outline of the proposed garment retargeting method.	30
4.3	Results from our method for retargeting 3D garment onto SMPL body meshes of differ-	
	ent poses and shapes (a) - (f); and on non-parametric 3D human scans (g) (h)	32
4.4	Results of real garments draped on unseen pose and shape	35
4.5	Wrinkle generation module architecture.	37

LIST OF FIGURES

4.6	Inference on T-shirt.	37
4.7	The figure shows different parametric cloth setting both tops and bottoms draped onto	
	SMPL extracted from the AMAAS Dataset.	39
4.8	Results of Laplace detail integration.	40
4.9	Comparison between two proposed wrinkle generation modules.	41
4.10	Retargetting 3D garments from CLOTH3D dataset onto non-parametric human bodies	
	from THumans2.0[68] dataset. Our approach can deal with layered clothing as well.	42
4.11	This shows the propogation of Isoembeddings from intersection points to the whole body.	42
4.12	The figure shows different real scanned garments of our Dress Me Up dataset draped	
	onto SMPLs of AMAAS dataset	43
4.13	Comparison of our method with M3DVTON[69] for draping non-parametric garments.	
	M3DVTON introduces false garment geometry (the sleeve of the t-shirt mapped to the	
	sleeveless part of the target geometry) to inaccurate geometries.	46
4.14	Qualitative results of our garment retargeting method on non-parametric avatars recon-	
	structed from internet images.	48
4.15	Comparison with M3DVTON.	49
4.16	Joint Masks	49
4.17	Comparison with DIG.	50
4.18	(a) Jump and (b) Smooth trajectory.	50
4.19	<i>Topwear:</i> The figure shows visualization of our collected dataset, first three rows depict	
	the geometry of our collected garment in different poses, while last three shows the	
	textured rendering of the respective geometries.	53
4.20	BottomWear: The figure shows visualization of our collected dataset, first three rows	
	depict the geometry of our collected garment in different poses, while last three shows	
	the textured rendering of the respective geometries.	54
4.21	The figure shows different real scanned garments of our Dress Me Up dataset draped	
	onto real-scans of T-humans2.0 human body scans, (a) shows the Dress Me Up's real-	
	garments and columns (b) and (d) show scanned humans of Thumans2.0, we employ	
	our proposed framework to drape these real garments to arbitrary real body scans of	
	Thumans2.0 dataset as visualized in columns (c) and (e).	56

Chapter 1

Introduction

Computer Vision is a field of research focused on enabling computers to understand and analyze the 3D world that is tantamount to human perception, if not better. Some of the most challenging problems tackled by this field are object detection, video tracking, 3d scene reconstruction and understanding, human pose estimation and modeling etc. Computer Vision also finds application in various domains like medical image analysis, animation, VFX in the entertainment industry, autonomous driving, robot planning and navigation, Virtual and Augmented Reality (VR/AR) and many more.



Figure 1.1: Different applications that use Metaverse.

In particular, Metaverse has gained massive popularity these days, with tech giants investing heavily in its research and development ^{1 2}. Metaverse is a virtual shared space combining physical and digital reality, and has numerous applications in various industries and research fields. As humans are three-dimensional beings, navigating and interacting in 3D through these virtual worlds is easier and more appealing than with a 2D screen. Hence Metaverse finds applications in like health sector, online social interaction, gaming and entertainment, education, E-Commerce transactions etc Figure 1.1. The recent development in the hardware is helpful in making AR/VR devices more accessible to customers. Computer vision and Artificial Intelligence (AI) play a vital role in the development and progress of Metaverse. Some of the challenges to be addressed in building a Metaverse including human-computer interactions, and scene understanding, rely on AI techniques for solutions. However, most of these methods rely on supervised learning that requires labor intense annotated data and have limited generalizability.

In this thesis, we aim to explore self-supervised methods in 3D computer vision that overcome the limitations of supervised methods. These methods bypass the need for annotated data, thus allowing to leverage a large corpus of data for training. In particular, we focus on self-supervised methods for UV parameterization of general objects and garment retargeting. In case of the former, UV parameterization refers to the mapping of 3D objects onto 2D planes. These flattened maps are useful for the efficient storage of data associated with a 3D model, such as texture, normals, albedo, depth etc. These texture maps facilitate easy manipulation of the 3D object appearance by editing the 2D texture maps Figure 1.2. As a later part, we address the problem of virtual try-on. This problem has got a lot of traction in the last couple of years owing to the pandemic, resulting in people shifting to the mode of online shopping. 3D virtual try-on allows users to virtually try on or test out different garments and visualize how they would look or fit on themselves without physically trying them on. It provides a simulated experience of trying on different styles, sizes, colors, or combinations before making a purchase decision. This can be extremely immersive especially in a VR setup. However, it involves addressing various challenges such as generalization to different poses and shapes of people in the image, non-rigid deformations of garments, generalization to a plethora of garments styles etc. Addressing these challenges can take the virtual try-on experience to a new level.

1.1 Motivation

Most of the applications of computer vision rely on Supervised Learning methods that have proven to be powerful tools and are widely used. However, they have several limitations ³. They generally require a large amount of labeled data for training, which are difficult to acquire, and labeling them can be time-consuming, expensive, or even infeasible in certain cases, especially when dealing with rare

¹https://finance.yahoo.com/news/tech-companies-pouring-billions-vr-175131376.html

²https://www.tomsguide.com/news/apple-vr-and-mixed-reality-headset-release-date-price-specs-and-leaks

³https://medium.com/analytics-vidhya/the-severe-limitations-of-supervised-learning-are-piling-up-ecalecf3e113



Figure 1.2: Appearance manipulation of mesh via texture editing.

events or niche domains. The training data needs to be representative of the target population covering a wide range of scenarios. Incorrect or mislabeled examples can adversely affect the model's performance leading to incorrect predictions. Supervised models primarily learn statistical patterns without a deep understanding of the underlying context. Hence, they may struggle to generalize well to unseen data that differs significantly from the training data and can be sensitive to noise or errors in the labeled data Figure 1.3. Overfitting can also occur when a supervised model becomes too complex and starts to memorize the training data instead of learning general patterns, resulting in poor performance on new, unseen data.

Self-supervised learning methods offer a viable alternative as they overcome several of the abovementioned limitations ⁴. These methods can leverage large amounts of readily available unlabeled data, which is often more abundant and easier to collect compared to labeled data. This reduces the reliance on costly and time-consuming data annotation processes. They also allow models to actively explore and understand unlabeled data, potentially discovering new patterns or novel aspects of the data that may not be captured by pre-defined labeled classes. These methods can learn to capture temporal or spatial relationships within data, modeling contextual information, which is valuable for several computer vision tasks. Hence, these methods can be useful in addressing the challenges of building the Metaverse.

Metaverse is expected to be the future universe, with economic projections reaching 800 billion by 2025 and 2.5 trillion by 2030⁵. In the Metaverse, users can interact with each other and a digital environment through avatars, which are computer-generated characters designed to resemble real human beings. Human modeling is used in the Metaverse to create these digital avatars. Using UV parameter-ized maps of avatars, textures can be accurately applied, resulting in realistic skin, clothing, and other visual details. This allows for the customization and personalization of avatars, enhancing their visual fidelity in the Metaverse. UV parameterization also plays a crucial role in the creation of virtual fashion and design within the Metaverse. Designers can create 2D patterns that are then mapped onto the UV coordinates of a human model, allowing for virtual clothing customization and fitting.

⁴https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/

⁵https://www.plugxr.com/augmented-reality/what-to-expect-in-metaverse-future-2030/





Character modeling in general, plays a significant role in the creation of animated films, television shows, and video games. Artists use specialized software like Autodesk Maya⁶, ZBrush⁷, or Blender⁸ to create 3D models of characters. These models serve as the foundation for rigging, animation, and rendering processes. UV parameterization plays a crucial role in character modeling. It can be beneficial for rigging and animation purposes as well. When a character model is rigged with a skeleton and control points for animation, UV parameterization can help preserve texture continuity during deformation. It also allows artists to apply textures, such as color, detail, and surface properties, onto a character model. By carefully laying out UV coordinates, animators can ensure that textures remain aligned and undistorted as the model moves, resulting in seamless and visually appealing animations.

Before mapping textures, the 3D model needs to be unwrapped, which involves flattening the model's surface into a 2D representation. This process involves cutting and rearranging the model's geometry to minimize distortions and optimize the use of texture space. Several such mapping tasks often include human priors that cannot be represented concisely through analytical methods, or else require computationally expensive numerical optimization at runtime. Hence, relying on data-driven methods while using modern statistical learning for semantic priors, can help approximate optimized solutions in a single forward pass and have a wide impact. Potential applications encompass tasks like 3D shape alignment to desired target, inferring high quality texture maps based on examples by artists, and seam-

⁶https://www.autodesk.in/

⁷https://www.maxon.net/en/zbrush

⁸https://www.blender.org/



Figure 1.4: Example of virtual garment try-on system.

less integration with optimization tasks like physics simulation. This also has the benefit of directly integrating into the graphics pipeline without the need to convert back and forth between different representations. It seems logical to harness deep-neural networks which have proven to be very effective in complex tasks, to achieve accurate and high-quality results.

The E-Commerce industry, fashion and apparel sectors in particular, have experienced tremendous growth in recent years, influenced by several factors such as advancements in technology, increasing internet penetration, changing consumer behavior, and the convenience it offers. Fashion and apparel e-commerce is expected to continue growing, with estimates that global online fashion sales will reach \$1.2 trillion by 2027 ⁹. The Metaverse can provide immersive virtual environments where users can explore and interact with products in a more realistic and engaging manner than traditional online shopping. Virtual reality (VR) and augmented reality (AR) technologies can allow shoppers to visualize and try on products virtually, enhancing the online shopping experience Figure 1.4. They can significantly reduce the need for customers to return products due to fit or appearance issues. Virtual try-on is a method to accurately simulate how the garment would look on a given person in terms of texture, color, fit, and overall appearance. Viewing from multiple views and 360⁰ visualization is another desired property of a virtual try-on system. Given these requirements, it seems optimal to design this virtual try-on in a 3D setting instead of 2D. Self-supervised methods can be used the leverage the available data to model a suitable system for 3D virtual try-on.

⁹https://www.statista.com/topics/9288/fashion-e-commerce-worldwide/#topicOverview

1.2 Problem Statement

As discussed, given the advantages of self-supervised methods, we address the problem of UV parameterization of general objects and 3D garment retargeting using these self-supervised learning approaches. However, using neural networks for inferring on meshes is, to a large extent, an open problem. Also, learning on 3D surfaces has fundamental issues that make it a harder problem than learning on 2D images. As opposed to images that have a fixed grid structure, 3D surfaces have significant geometric and topological variation. A pair of surfaces representing the same geometric shape can have different discretization making it difficult to establish an association between them. The network should also be able to account for the lack of ordering of points as well as have equivariance to different spatial orientations of the 3D surface. Thus suitable architectures need to be designed to overcome these challenges. Some methods choose to work with one fixed triangulation which limits their applications to real-world data where the triangulation is not known in advance. Hence, the network needs to be designed to capture the implicit features of the geometric shape. Keeping these challenges in mind, we wish to address the following two problem statements.



Figure 1.5: The objective is to achieve UV parameterisation and Garment Retargeting.

1.2.1 UV parameterisation

First, we focus on the problem of surface parameterization of arbitrary 3D objects with both closed and open surfaces via a self-supervised framework. The existing methods for surface parameterization of arbitrary 3D objects face challenges when dealing with closed surfaces and regions of extreme extrinsic curvature. We aim to develop a novel, self-supervised framework that addresses these issues and enables the efficient and accurate surface parameterization of arbitrary 3D objects. Mapping a surface from 3D to 2D almost always introduces a certain amount of distortion, and the aim is to keep this distortion as low as possible. Finding optimal seams that lead to low distortion is another challenge. The objective of this research is to propose a learning-based approach that partitions closed surfaces into open patches and independently parameterizes them. Additionally, the framework should learn the surface parameterization of open 3D surfaces to a UV plane. The solution should enforce meaningful UV mapping and achieve desired properties such as isometric, conformal, and area-preserving parameterizations. The framework should leverage learning-based methods to provide a multi-scale characterization of the underlying surface, ensuring a global-to-local context for each vertex. Furthermore, the proposed solution should be discretization agnostic, allow for learning on lower-resolution meshes and enable the direct inference of parameterization for high-resolution meshes without the need for retraining.

1.2.2 Garment Retargeting

Next, we aim to innovate a self-supervised method that can retarget real, non-parametric garment meshes over a target human body (either parametric or non-parametric). This 3D virtual try-on system needs to generalize to arbitrary body shapes and poses, modeling topological differences among various categories of garments, along with realistic deformations arising out of the physical interaction with the underlying body and resolving the penetration/intersection of the garment with the underlying body. These systems need to run in real time with very less delay. Given a 3D garment mesh and a target 3D human mesh, the aim is to first estimate correspondences between the two meshes using a novel representation, which provides an initial placement of the garment around the target body as a coarse retargeting initialization. Further, realistic garment deformation details need to be added, either via physics-based simulation methods or via data-driven methods. Unlike existing methods that rely on skinning, we aim to repose any arbitrary non-parametric garment on any parametric or non-parametric target body. Finally, as a post-processing step, fine wrinkles and details can be introduced in the retargeted garment conditioned on the pose and shape of the target human body. Additionally, due to the lack of any real-world datasets for 3D garment retargeting, we aim to curate our own dataset, containing different garments worn by multiple subjects in arbitrary poses. This data can serve as ground truth for the evaluation of retargeting methods.

1.3 Research Landscape

1.3.1 UV parameterisation

Early methods for mesh parameterization were based on optimization and can be categorized as single patch, free boundary methods, which include computing a piecewise linear least square solution of the Cauchy-Reimann equations. One method minimized the Dirichlet energy of the flattened area. A line of methods aimed at optimizing the angles between the flattened mesh and the input mesh. Another category of parameterization is single-patch fixed boundary methods that explicitly incorporate boundary conditions. The above-mentioned categories require the mesh to be homogeneous to a disk topology. However, global parameterization methods can deal with meshes of arbitrary genus. They

achieve this by partitioning the closed mesh into multiple open patches or detecting one or more seams to cut the mesh, making it homogenous to a disk. A class of methods jointly solve for optimal seams and parameterization. Neural parameterization methods have gained popularity over the past few years due to their ability to address ill-posed problems. AtlasNet [20], and DGP [63] propose a way of surface reconstruction and parameterization by training a neural network to represent a single UV chart over the reconstructed surface. Both methods use a fixed number of patches for the surface parameterization but require a different neural network for every patch, which is overkill and difficult to scale. A recent work AUV-Net [12] takes a point cloud as input and learns parameterization of aligned surfaces (e.g., faces and humans in T-poses) using a cycle-loss, but requires all the meshes to have similar topology and same orientation to enable learning. Moreover, the proposed two-patch estimation method is very naive and cannot scale to an arbitrary number of patches. Another recent method [1] learns intrinsic mapping of arbitrary surfaces in a supervised setup, where a conventional method acts as the ground truth. However, no method exists that achieve this in a self-supervised setup.

1.3.2 Garment Retargeting

The current solutions for 3D garment retargeting can be broadly categorized into two types: traditional graphics simulation pipelines and modern deep learning techniques. The simulation-based approaches are known for their ability to depict deformations and wrinkles in garments accurately. However, they often depend on previous frames of reference to determine velocity and acceleration parameters. This reliance on trajectory information poses a challenge in AR/VR applications where fashion shoppers want to see how a garment would appear on their digital avatars without having to provide a specific trajectory. To overcome this limitation, an alternative approach is needed, namely an arbitrary pose and shape re-targeting technique. Recent advancements in deep learning have made progress in this direction by using supervised training strategies. These techniques learn the skinning weights of a parametric garment to drape it onto a parametric human body, with the popular choice being the SMPL (Skinned Multi-Person Linear) [38] body model. The garments themselves are derived from the SMPL body mesh. However, there are several issues associated with these approaches. Firstly, they require a large number of training samples, which are generated by simulating garments on top of the body mesh using software like Blender. These parametric garments are synthetic and do not accurately represent garments with diverse topologies. Furthermore, real scans or reconstructed garments from images are not inherently parametric, making it challenging for the existing methods to handle them.

1.4 Contribution

1. **Self-Supervised setup for UV parameterization:** We present a novel self-supervised framework for learning the discretization-agnostic surface parameterization of arbitrary 3D objects with both closed and open surfaces. We evaluate our framework on multiple 3D objects from the publicly

available SHREC [Lian et al. 2011] dataset and report superior/faster UV parameterization over conventional methods.

2. 3D virtual Try-on System: We present a self-supervised framework for retargeting real, non-parametric 3D garments on an arbitrary target human body. Our novel formulation can retarget in-the-wild, arbitrary non-parametric garments over parametric as well as non-parametric bodies in arbitrary pose and shape. We propose a first-of-its-kind real-world 3D garment dataset for evaluating our proposed framework.

1.5 Thesis Roadmap

In this chapter, we introduced motivation for self-supervised methods, our problem setup, challenges associated with it, limitations of existing methods, and our contributions.

In *Chapter-2*, we provide the necessary background for this thesis and briefly summarize the aspects of various ways to represent 3D data. We provide a clear idea of UV and texture mapping and distinction between vertex-colored vs textured reconstruction. We also discuss about Skinned Multi Person Linear model (SMPL) - a parametric human body model which can act as a useful prior.

Chapter-3 focuses on a solution for self-supervised methods for UV parameterization of arbitrary meshes. *Chapter-4* discusses a robust framework for a realistic garment retargeting system in real-time.

Finally, Chapter-5 lays down a discussion of the impact of the proposed methods, it's limitations as well as a future direction for research.

Chapter 2

Background

In this chapter, we provide a brief introduction to relevant concepts and terminologies used in the thesis. First, we'll be discussing the representation of 3D geometry. Next, we briefly discuss parameterisation methods and evaluation metric used for the same. We also give a brief introduction to SMPL [38] representation which is a parametric human body model, used in several cases to either represent a human body or used as a body prior.

2.1 3D Representations

There exist several representations for digital representation of a 3D shape such as point cloud, polygonal mesh, voxel grid, implicit surfaces, SDF(sign distance functions) etc. We'll be discussing only about the point cloud and mesh as they're are relevent for the thesis Figure 2.1.



Figure 2.1: 3D representations as point cloud and mesh.

2.1.1 Point Cloud

Point cloud represents objects or scenes as a set of discrete points in 3D space. Each point in the cloud corresponds to a specific location in 3D Cartesian coordinates and may be associated with additional attributes such as color, intensity, or surface normals. In a point cloud with N number of points, each point $P_i \in \mathbb{R}^3$ is represented as $P_i = \{x, y, z\}$ with i = (1, 2, ..., N). The points in a point cloud are often obtained through 3D scanning techniques or sensors like LiDAR (Light Detection and Ranging), structured light scanners, or photogrammetry. They are typically considered as an unordered set, lacking any inherent connectivity information. Point clouds can be converted to mesh for easier visualisation, rendering and further processing of data. This is achieved using methods like Poisson surface reconstruction, marching cubes, delaunay triangulation, ball pivoting algorithm etc.

2.1.2 Mesh

The mesh representation, also referred to as a polygon mesh, is a commonly employed data structure used to depict the surface geometry of 3D objects or scenes. It offers a discrete approximation of the continuous surface by subdividing it into a set of interconnected polygons. It is composed of vertices, edges and faces, which together define the shape or topology of the object. A face can have three or more vertices. A mesh with three vertices per face is called a triangular mesh and four vertices is called a quad mesh. *OBJ* is one of the common file format used for storing the polygonal meshes and stores different elements in the following format:

- Vertices: (x, y, z) coordinates of the points. Vertex color can also be stored per point. Each vertex in .obj file start with "v". (e.g. v -1.000000 1.000000 -1.000000 255 255 255).
- Texture Coordinates: Each vertex is assigned a two dimensional texture coordinate for texture mapping. Each coordinate is represented by "vt" (e.g. vt 0.250000 0.250000).
- Vertex Normals: The normal per vertex, (n_x, n_y, n_z) are represented as vn 1.000000 1.000000, 1.000000.
- Faces: A face in a .obj file stores the indices of the vertices that together form a given face and the normal vector and texture coordinate of those vertices. A sample of a face is represented as "f 27/1/43 14/3/46 42/11/87". *f* is used to declare a face, followed by three groups of three numbers, seperated by a '/'. The first number in each group defines the index of the vertex in the vertex array. The second number is the index of the textrue coordinate per vertex and the third number is the index of the vertex normal. The arrays are 1-based in OBJ file format, i.e. first element in the array has index 1).



Figure 2.2: Parameterisation of a mesh.

2.2 Parameterisation

As mentioned earlier, parameterisation refers to one-to-one mapping of suitable domain, which in our case is a 3D mesh to a 2D surface. This has many application in various fields of science and engineering. However, the main motivation for developing the first parameterisation methods was due its application in texture mapping for better visual quality of 3D models. Parameterisation is achieved by flatening or unwarping of a 3D mesh onto a 2D plane. The simple example of unwrapping a cube is shown in Figure 2.2(b). Once this is done, each face of the mesh can be associated with the texture information by overlaying the texture image (called texture map or UV map) onto the parameterised 2D map of the mesh. This process is shown in Figure 2.2(a).

2.3 SMPL: Skinned Multi-Person Linear model

The primary goal of SMPL [38] is to provide a compact and efficient representation of the human body that can be easily manipulated and animated. It allows for the generation of realistic and controllable human characters, which is crucial in various applications such as animation, virtual reality, gaming, and biomechanical simulations. To construct the SMPL model, a large-scale dataset of 3D body scans was used to learn a statistical model of human body shape and pose variations. Using this dataset, a statistical analysis was performed to derive the low-dimensional shape and pose spaces that are used in the SMPL model. Separate models are available for men and women.

The SMPL model consists of two main components: shape parameters $\vec{\beta} \in \mathbb{R}^{3K}$ and pose parameters $\vec{\theta} \in \mathbb{R}^{3K}$. The shape parameters control the overall body shape and are represented as a low-dimensional vector. By adjusting these parameters, it is possible to generate a wide variety of body shapes, ranging from thin to obese individuals. The pose parameters control the articulation of the body joints. They describe the rotation angles of the different body parts, such as the arms, legs, and torso. By manipulating these parameters, it is possible to animate the SMPL model, bringing it to life with a wide range of movements and poses. Thus SMPL enables the creation of diverse characters with different body types.

Different blend shapes for identity, pose, and soft-tissue dynamics are additively combined with a rest template $T \in \mathbb{R}^{3N}$ before being transformed by blend skinning.



Figure 2.3: (a) Template mesh with blend weights indicated by color and joints shown in white. (b) With identity-driven blendshape contribution only (c) With the addition of of pose blend shapes in preparation for the split pose; note the expansion of the hips. (d) Deformed vertices reposed by dual quaternion skinning for the split pose. Figure adopted from [38]

A single SMPL model is composed of N = 6890 vertices and K = 27 joints. The following notations are used:

- Blend shape function: $B_S(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \to \mathbb{R}^{3N}$ (This takes as input the shape parameters($\vec{\beta}$) and outputs a blend shape sculpting the subject identity.)
- Joint regressor function: $J(\vec{\beta}) : \mathbb{R}^{|\vec{\beta}|} \to \mathbb{R}^{3N}$
- Pose-dependent blend shape function: $B_P(\vec{\theta}) : \mathbb{R}^{|\vec{\theta}|} \to \mathbb{R}^{3N}$ (This takes as input pose parameters $(\vec{\theta})$ and accounts for pose dependent deformations.)
- Blend skinning function: W(.) (Linear or Dual-quaternion)

The pose of the body is defined by the standard skeleton rig where $\vec{\omega}_k \in \mathbb{R}^3$ denotes the axis angle representation with respect to its parent in the Kinematic tree. The SMPL rig has K = 23 joints, hence a pose $\vec{\theta} = [\vec{\omega}_0^T, \dots, \vec{\omega}_K^T]^T$ defined by $|\vec{\theta}| = 3 \times 23 + 3 = 72$ parameters; 3 for each joints and 3 for root orientation. In blend skinning, we attach the surface of a mesh to an underlying skeletal structure. Each vertex in the mesh surface is transformed using a weighted influence of its neighboring bones. This influence can be defined linearly as in Linear Blend Skinning (LBS).

The process of deformation is shown in Figure 2.3. Then template mesh T and the blend weights W are plotted in (a). The change in template with shape parameters β is depicted in (b). (c) shows the deformations in the template mesh according to change in pose θ . (d) shows the final mesh with new joint locations and deformed vertex obtained by applying blend skinning on the previous mesh (c). The

new position of the vertices are obtained via the following formula:

$$t'_{i} = \sum_{k=1}^{K} w_{k,i} G'_{k}(\theta, J(\beta))(t_{i} + b_{S,i}(\beta) + b_{P,i}(\theta))$$
(2.1)

where $G_k(\theta, J)$ is world transformation of joint k and $w_{k,i}$ is the weight associated with vertex i to joint K. $b_{S,i}(\beta)$ and $b_{P,i}(\theta)$ is vertex i in $B_S(\beta)$ and $B_P(\theta)$ respectively.

In conclusion, SMPL is a powerful parametric model for representing human body shape and pose. It has become a cornerstone in computer graphics and computer vision, enabling the generation of realistic and controllable human characters. Its simplicity, efficiency, and flexibility make it a valuable tool in a wide range of applications.

Chapter 3

Discretization-Agnostic Deep Self-Supervised 3D Surface Parameterization

As mentioned earlier, learning-based surface parameterization has several advantages. In this chapter, we discuss a self-supervised method for achieving this. We present a novel framework for learning the discretization-agnostic surface parameterization of arbitrary 3D objects with both open and closed surfaces. Our framework leverages diffusion-enabled global-to-local shape context for each vertex first to partition the closed surface into multiple patches using the proposed self-supervised PatchNet and subsequently perform independent UV parameterization of these patches by learning forward and backward UV mapping for individual patches. Thus, our framework enables learning a discretization agnostic parameterization at a lower resolution and then directly inferring the parameterization for a higherresolution mesh without retraining. We evaluate our framework on multiple 3D objects from the publicly available SHREC [35] dataset and report superior/faster UV parameterization over conventional methods.

3.1 Introduction

Estimating the UV parameterization of arbitrary 3D surfaces lies at the core of computer graphics and geometry processing domain, with a wide range of applications such as 3D modelling, texturemapping, remeshing, simulation, etc. Formally, it is defined as the projection of vertices of a tessellated surface (polygon mesh) onto a 2D map (UV plane). Determining the aforementioned mapping is not a trivial task and demands a solution with specific properties. The estimated mapping is expected to be isometric, conformal, and non-overlapping. Existing conventional methods [62, 34, 48, 51, 36] aim to estimate an object-centric mapping with an iterative optimization process, focusing on minimizing an energy function explicitly constructed to retain the desired properties. However, they face scalability issues while dealing with high-resolution object meshes and are also prone to local minima.

With the advent of deep learning, researchers are harnessing the power of neural networks to solve various ill-posed problems, offering tractable solutions. Neural surface parameterization has recently



Figure 3.1: UV parameterization for open and closed surfaces estimated via our proposed framework.

been attempted [1] but under supervised, data-driven settings, requiring a large amount of training data. Such supervised learning solutions get subjected to data bias and hence suffer from poor generalization to unseen, out-of-distribution samples.

This paper presents a novel, self-supervised framework for learning the discretization-agnostic surface parameterization of arbitrary 3D objects with both open and closed surfaces as shown in Figure 4.1. First, to handle closed surfaces (e.g., a sphere) or surfaces with regions of extreme extrinsic curvature, we propose a learning-based partitioning of the given surface into multiple open patches, which are independently parameterized. To this end, we employ a self-supervised network that assigns each 3D point of the surface to one of the patches, trained using losses based on local features (such as face-normals) and geodesic relationships within the patch.

Subsequently, we propose to learn the surface parameterization of an arbitrary (open) 3D surface to a UV plane using a *Multi-layer Perceptron* (MLP). More specifically, given a *open* 3D surface (patch), we train the forward MLP to predict per-point UV coordinates independently. In order to ensure a meaningful UV mapping, we enforce cycle-consistency loss between the input and reconstructed surface by learning a backward mapping (UV-to-3D) MLP. Additional losses are employed to achieve desired properties of surface parameterization, i.e., isometric, conformal, and area-preserving. A diffusion process [53] over the mesh provides a multi-scale characterization of the underlying surface, entailing a global-to-local context for each vertex. Hence, the DiffusionNet backbone is used for Patch-Net, and similarly, respective features are appended while learning surface parameterization to achieve discretization-agnostic UV mapping. A key advantage of learning a discretization-agnostic parameterization for high resolution meshes at a lower resolution and then directly infer the parameterization for high resolution meshes without retraining, as shown in Figure 3.3.

3.2 Background

Conventional Methods for Surface Parameterization: Conventional methods to solve mesh parameterization generally fall into one of the three categories. The first one is *single-patch, fixed boundary* methods, e.g. harmonic parameterization[62], which projects the boundary vertices onto a circle in UV space and computes two harmonic functions (one for u and one for v coordinate). LSCM[34], which is a *single-patch, free boundary* parameterization method, minimizes the conformal (angular) distortion. Unlike harmonic parameterization, it does not need to have a fixed boundary. Both the aforementioned categories can only deal with open surfaces with genus 0. The third category is formally known as *global parameterization* method, which can deal with meshes of arbitrary genus. They achieve this by cutting the given mesh into the patch(es) and individually parameterizing each patch. The generated per-patch maps are discontinuous around the cut when laid down in the UV space. This discontinuity can be seen as seams on the 3D surface. Another class of global methods try to detect one or more seams to cut the mesh to make it open and then parameterize it. OptCuts[36] and Boundary-First Flattening[51] fall into this category. There are global seamless parameterization methods as well, but they are out of the scope of this work.

Neural Methods for Surface Parameterization: Neural parameterization methods have gained popularity in the past few years due to advancements in deep learning methodologies and hardware stack. AtlasNet[20] was one of the first works along these lines, which tries to generalize on different classes of objects. However, its use case was directed more towards surface reconstruction than surface parameterization. Another method, DGP[63], builds upon AtlasNet and proposes an object-centric way of surface reconstruction by overfitting a neural network representing a local chart parameterization. Both methods use a fixed number of patches for the surface parameterization but require a different neural network for every patch, which is overkill and difficult to scale. Another work, AUV-Net[12], takes a point cloud as input and learns parameterization of aligned surfaces (e.g., faces and humans in T-poses) using a cycle-loss and smoothness loss. However they require all the geometries of the same category and in the same orientation. Moreover, their patch estimation method is very naive and can not scale to an arbitrary number of patches. All the aforementioned learning-based methods sample points in the UV space and learn to map it to a 3D surface, thereby assuming the UV space itself, hence failing to produce a plausible UV map. Another very recent method [1] learns intrinsic mapping of arbitrary surfaces in a supervised fashion where a conventional method acts as the ground truth.

3.3 Method

We now describe the proposed framework in detail. The input to our framework is a mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{F}, \mathcal{N}_V\}$, where \mathcal{V}, \mathcal{F} and $\mathcal{N}_{\mathcal{V}}$ are the sets of vertex positions, faces and vertex-normals respectively. Our framework consists of two modules: (*i*) Patch extraction module and (*ii*) Surface parameterization module.



Figure 3.2: The outline of proposed framework.

3.3.1 Patch Extraction Module

Handling surfaces with regions of high extrinsic curvature or closed topology requires the 3D manifold to be partitioned into multiple open patches to minimize distortion and overlap. Each patch is defined as $\mathcal{P}_k = \{\mathcal{V}_k, \mathcal{F}_k, \mathcal{N}_{Fk}\}$ (k = 1, 2, ..., K), where $\mathcal{V}_k \subseteq \mathcal{V}$ is the set of vertices belonging to \mathcal{P}_k . $\mathcal{F}_k \subseteq \mathcal{F}$ is the set of faces defined on \mathcal{V}_k and $\mathcal{N}_{Fk} \subseteq \mathcal{N}_F$ is the associated set of face-normals. We propose PatchNet with parameters ϕ_{patch} , which learns to assign each vertex of \mathcal{M} to one of the Kpatches, as shown in Figure 3.2. Here, K is a controllable parameter and can vary based on the acceptable amount of distortion in the input mesh. To learn the parameters ϕ_{patch} , we minimize the following cosine similarity constraint on the estimated patches:

$$\mathcal{L}_{cos} = \sum_{k=1}^{K} \frac{1}{|\mathcal{F}_k|} \left[1 - \left(\sum_{i,j \in \mathcal{F}_k} (\hat{n}_i^T \hat{n}_j) \right) \right]^2$$
(3.1)

Trained network on low resolution mesh



Figure 3.3: Discretization-agnostic UV parameterization.

where $i, j \in \mathcal{F}_k$ are the pair of faces with unit normal vectors $\hat{n}_i, \hat{n}_j \in \mathcal{N}_{Fk}$, respectively, and $|\mathcal{F}_k|$ is the number of faces in that patch. The above constraint has the effect of producing locally flat patches. However, geodesically far-apart triangles with high cosine similarity may be assigned to the same patch, which is undesirable. To circumvent such disjoint assignments, we minimize the following additional constraint:

$$\mathcal{L}_{geo} = \sum_{k=1}^{K} \frac{1}{|\mathcal{P}_k|} \left(\sum_{i,j \in \mathcal{V}_k} g(i,j) \right)$$
(3.2)

where g(i, j) denotes the geodesic distance between the pair of vertices i & j within the patch and $|\mathcal{P}_k|$ is the number of vertices in that patch. We model PatchNet using DiffusionNet [53] architecture to achieve multi-scale characterization of the underlying surface, entailing a global-to-local context for all the vertices. Input to PatchNet is the vertices \mathcal{V} and vertex-normals \mathcal{N}_V , and the output is the predicted assignment probability for all the vertices to each of the K patches. Subsequently, per-face probabilities are obtained by taking the mean probabilities of the corresponding face vertices. We further consolidate the per-face probabilities by taking an average over neighboring faces, and then each face is assigned to the patch with the highest probability. Note that the whole mesh can be considered as a single patch in the case of a open surface with extrinsic curvature of low variability. The combined objective function for patch extraction becomes $\mathcal{L}_{patch} = \lambda_{cos}\mathcal{L}_{cos} + \lambda_{geo}\mathcal{L}_{geo}$.



Figure 3.4: Comparison of error plots for QCE and ASE with other methods. First two categories (a) Bird, (b) Pliers are taken from SHREC dataset; (c) Armadillo & (d) Spot.

3.3.2 Surface Parameterization Module

Each patch $\mathcal{P}_k = \{\mathcal{V}_k, \mathcal{F}_k, \mathcal{N}_k\}$ is treated as a separate open surface and is independently parameterized. Let $f : \mathbb{R}^3 \to \mathbb{R}^2$ be the mapping of each vertex $v \in \mathcal{V}_k$ to a 2D point u on the UV plane. We propose to represent f using a *forward* mapping network MLP_f with learnable parameters ϕ_f . First, the set of vertices \mathcal{V}_k for the given patch is passed to the diffusion block to get a global shape encoding $\psi \in \mathbb{R}^{128}$. Per-vertex input given to MLP_f is $z \in \mathbb{R}^{131}$ (v concatenated with ψ) and the output is $u \in \mathbb{R}^2$ (UV coordinate), i.e. $u = MLP_f(z)$. Since we do not have corresponding ground truth UV coordinates, we resort to a self-supervised cycle-consistency loss. We employ another $MLP_{f^{-1}}$ with learnable parameters $\phi_{f^{-1}}$ to represent the *backward* mapping $f^{-1} : \mathbb{R}^2 \to \mathbb{R}^3$. $MLP_{f^{-1}}$ takes uas input and predicts its corresponding 3D position, which ideally should match with the input vertex position v. We enforce this consistency by minimizing the following cycle loss:

$$\mathcal{L}_{cycle} = \frac{1}{|\mathcal{V}_k|} \sum_{v \in \mathcal{V}_k} \left(v - MLP_{f^{-1}}(u) \right)^2$$
(3.3)

Note that due to presence of non-linear activation functions in MLP_f and $MLP_{f^{-1}}$, the condition $\phi_f \cdot \phi_{f^{-1}} = I$ need not hold. Per-vertex prediction can be noisy, resulting in an irregular UV space. Conditioning the MLPs with the diffusion-based global shape-encoding ψ regularizes the UV prediction and improves the output of $MLP_{f^{-1}}$. We further add losses to enforce desired properties of surface



Figure 3.5: Additional Qualititative Results

parameterization, namely, L_{iso} provides isometric behaviour, L_{angle} preserves angles of the faces and L_{area} preserves face-area (neglecting uniform scaling). The final objective function for surface parameterization is given follows:

$$\mathcal{L}_{uv} = \lambda_1 \mathcal{L}_{cycle} + \lambda_2 \mathcal{L}_{iso} + \lambda_3 \mathcal{L}_{angle} + \lambda_4 \mathcal{L}_{area}.$$
(3.4)

3.3.3 Losses

The final objective function for surface parameterization is given as:

$$\mathcal{L}_{param} = \lambda_1 \mathcal{L}_{cycle} + \lambda_2 \mathcal{L}_{iso} + \lambda_3 \mathcal{L}_{angle} + \lambda_4 \mathcal{L}_{area}$$
(3.5)

The cycle consistency loss L_{cycle} imposes bijectivity constaints in the UV space, while the isometric loss L_{iso} imposes isometricity. The isometric loss L_{iso} is designed to impose isometric constraint in the UV space. Inspired by [75], the loss ensures that the geodesic distance between a pair of vertices in 3D space $G_d \in \mathbb{R}^{V \times V}$ is equal to the euclidean distance $E_d \in \mathbb{R}^{V \times V}$ in the UV space. The L_{iso} is given as:

$$\mathcal{L}_{iso} = ||G_d, E_d|| \tag{3.6}$$

	BFF		0	ptCuts	Ours		
Class	QCE↓	—ASE—↓	QCE↓	—ASE—↓	QCE↓	—ASE—↓	
Laptop	1.046	2.052	1.045	2.005	1.196	2.420	
Pliers	1.112	1.909	1.128	1.391	1.274	2.895	
Rabbit	1.132	2.116	1.160	2.062	1.183	0.992	
Scissors	1.156	1.456	1.122	1.276	1.261	2.728	
Bird	2.130	1.103	1.129	1.928	1.262	1.996	

Table 3.1: Comparison of QCE and ASE metrics with BFF [51] and OptCuts [36] on SHREC dataset.

where ||.|| represents the L2 norm. This loss is imposed only on geodesic distances less than a certain threshold σ . We choose $\sigma = 0.2$ for all our experiments.

We use L_{angle} loss to reduce conformal error in the UV space. We take an L2 norm between the angles $\theta_{i=1}^3$ per face belonging to F in the 3D space and faces f in UV space, given as:

$$\mathcal{L}_{angle} = \frac{1}{|F|} \sum_{j=1}^{j=|F|} \frac{1}{3} \sum_{i=1}^{3} \left(\cos\left(f_{\theta_i}^j\right) - \cos\left(F_{\theta_i}^j\right) \right)^2$$
(3.7)

where |F| is total the number of faces.

Similarly, L_{area} loss is used to minimise the area distortion by taking an L2 norm between the areas a_p, a_q of the faces f in the 3D space and faces F in UV space, respectively. The loss is given as follows:

$$\mathcal{L}_{area} = \frac{1}{|F|} \sum_{a_p, a_q \in f, F} \left(a_p, a_q \right)^2$$
(3.8)

3.4 Experiments and Results

3.4.1 Implementation Details

For PatchNet, we use DiffusionNet[53] architecture with 4-blocks, channel width of 128 and 64 eigenbasis vectors for spectral acceleration. We use ReLU activations at intermediate layers and softmax function after the final output layer. The surface parameterization module uses an 8 layer MLP with 1.3×10^6 parameters for both forward and backward MLP with LeakyReLU activations in-between the layers and *tanh* at the final output layer. We use the PatchNet loss weights $\{\lambda_{cos}, \lambda_{geo}\} = [1.0, 1.0]$ and the parameterization loss weights $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} = [1.0, 1.0, 0.001, 0.001]$. We use ADAM optimizer with a learning rate of 10^{-3} and back size of 1 on a single RTX 2080Ti GPU for all our experiments.

The framework is implemented in PyTorch Lightning, trained on a single RTX 2080Ti GPU. We use xatlas¹ to pack the individual patches into the final UV atlas.

Resolution	BFF	OptCuts	Ours	Resolution	BFF	OptCuts	Ours
30K	17.41 sec	> 10 min	2.92 sec	40K	5 sec	4 sec	3.1 sec
100K	61.04 sec	$> 10 \min$	5.02 sec	100K	14 sec	12 sec	4.2 sec

Table 3.2: Comparison of computation time.

3.4.2 EVALUATION METRIC

(a) Stanford's Armadillo

We use Quasi Conformal Error (QCE) [49] and Area Scaling Error (ASE) [51] for evaluation of the UV distortions. QCE measures the angular distortion based on the ratio of the singular values of each face mapping. The ideal QCE value is 1, and a higher value implies distortion. ASE measures the scale factor of the mapped faces. Negative ASE values imply shrinkage; positives imply increase, and zero implies no area distortion in mapping.



Figure 3.6: Effect of geodesic loss.

Figure 3.7: Effect of DiffusionNet embeddings.

(b) Shark [35]

3.5 Additional Qualitative Results

Figure 3.5 shows qualitative results of our framework on arbitrary closed ((a)-(d)) as well as open meshes. The patches are extracted and parameterized individually for open meshes to form a UV atlas. In the case of (e), our method estimates a reliable surface parameterization even with high extrinsic curvature. Moreover, meshes (a), (b) & (c) are the unseen test samples from their respective classes, which are directly inferred. On the other hand, for meshes (d) & (e), parameterization is obtained by training a network till convergence. This shows that apart from learning parameterization in an

¹https://github.com/mworchel/xatlas-python



Figure 3.8: Patches are used to obtain multiple open surfaces from closed surfaces. As we increase the number of patches, the conformal and angular distortion gets reduced.

object-centric way, our framework can also generalize to a specific class/category and perform well on category-specific samples.

3.6 Results & Evaluation

We compute Quasi-Conformal Error (QCE) and Area Scale Error (ASE) on the final texture atlas for quantitative and qualitative evaluation. Please refer to the supplementary for their description. **Qualitative Comparison:** We compare our framework with BFF[51] and OptCuts[36] in Figure 3.4. As shown, our framework performs on par with these methods on varying geometrical shapes. **Quantitative Comparison:**

In Table 3.1, we compare our framework with BFF[51] and Opt-Cuts[36] on a few classes of SHREC [35] dataset using QCE and ASE metric. We train our network on 16 meshes for each mentioned class and compute errors on 4 test sample meshes. Please note that, instead of purely object-centric learning, we compare on a category-specific generalized network, and our performance is comparable to other object-centric methods. Such generalization can be attributed to intrinsic characterization encoded in diffusion features used in our surface parametrization module.

Discretization-agnostic Learning:

Figure 3.3 shows the discretization-agnostic learning capability of our framework. We train on a mesh with only $\sim 3K$ vertices and directly infer at high resolutions ($\sim 35K$ and $\sim 100K$ vertices). Please note that the error values for high-resolution meshes stay close to the low-resolution mesh, as observed in the error plots.


Figure 3.9: Ablation study of different losses.

Figure 3.10: Comparison of global and local embeddings.

More importantly, discretization-agnostic learning allows us to reduce the computation time significantly compared to other methods. Specifically, we train our method on the decimated mesh with $\sim 2K$ vertices and compare our computation time with other two methods at higher resolution as shown in Table 3.2.

3.7 Ablation Study

3.7.1 Patch Extraction Module

Effect of Geodesic Loss:

In section-3.1, we stated that the geodesically far-apart faces with high-cosine similarity might get assigned to the same patch, producing unwanted patches of extreme curvature. However, incorporating \mathcal{L}_{geo} into the objective function tends to sort out this issue as it penalizes the faces geodesically far apart and belonging to the same patch. This improvement is evident in Figure 3.6.

3.7.2 Surface Parameterization Module

Effect of DiffusionNet Embeddings:

As described in the method section of the main draft, the MLP_f and $MLP_{f^{-1}}$ take a global encoding ψ as input along with vertex position. This global encoding/embedding is the combination of the DiffusionNet features of all the vertices. Instead of using global encoding, per-vertex features from DiffusionNet can also be passed directly to the MLPs as input. However, we argue that per-vertex features are noisy and capture minimal global context, resulting in an irregular UV space and undesired UV coordinates. It can be observed in Figure 3.7 that when global encoding of DiffusionNet features is incorporated, the quasi-conformal error (QCE) drops. Moreover, from Figure 3.10, it is clear that the global encoding ψ provides the better context of the global shape as compared to per-vertex embeddings, thereby regularizing the UV space and hence producing better output both qualitatively and quantitatively (minimal overlapping and lower QCE value).

Effect of No of Patches on Parameterization:



Figure 3.11: Few limitations of current method.

Figure 3.8 shows the trade-off between distortion (value of QCE and ASE) and the number of patches. With the increase in the number of patches, the distortion follows an up and down curve but eventually reduces significantly.

3.7.3 Effect of Loss Functions:

The cycle loss \mathcal{L}_{cycle} is crucial for self-supervised training of the parameterization module. However, it is not sufficient to get desired properties (isometricity, conformality etc.). Figure 3.9 demonstrate the effect of additional loss functions, where the QCE value drops when \mathcal{L}_{iso} is introduced into the objective function, and it drops further when $\mathcal{L}_{angle} \& \mathcal{L}_{area}$ are also included.

3.8 Conclusion

We proposed a novel self-supervised learning-based framework for surface parameterization of open and closed surfaces. Our framework enables discretization-agnostic learning, enabling parameterization of meshes with arbitrary topology. We show significant improvement in inference time performance on high-resolution meshes. We also point out some of the limitations of this method in Figure 3.11. First, there might be some disconnected small patched after patch prediction. Even with our cycle consistency loss, some overlaps might still occur as there are no hard constraints to avoid them. In case of geometry with high extrinsic curvature, the network tends to flatten it as shown in the figure, which is undesirable. In such cases, more patches are required. These limitations can be overcome by designing more suitable losses.

Acknowledgement: We thank Dhawal Sirikonda for helping us with the visualization of the QCE error metric

Chapter 4

Dress Me Up: A Dataset and Method for Self-Supervised 3D Garment Retargeting



Figure 4.1: 3D garment retargeting on real human scans using our approach (left) and our real 3D garment dataset samples(right)

As discussed earlier, performing virtual try-on in 3D space indeed offers several advantages. However, given the limited amount of real-world garments available, we explore self-supervised methods to achieve this task. We discuss a method for draping non-parameterized, 3D garment meshes over human body meshes of arbitrary shapes and poses. SMPL body model is used to obtain Isomap Embedding based correspondences between the garment and the human body, to get a coarse alignment between the two meshes. We perform self-supervised refinement using PointNet embeddings of the coarsely aligned garment and the SMPL body model to improve the retargeting. We propose novel self-supervised losses which are used to train an MLP for solving physical interactions with the human body and resolving intersections between the meshes. Further, we propose wrinkle generation module to generate realistic wrinkles on the draped garment. We also contribute a new dataset of real-world reposed garments with realistic noise and topological deformations. This consists of garments captured using 7 Kinect Azure depth sensors, post-processed using multiview KinectFusion, followed by manual garment extraction. Some example of dataset is shown in Figure 4.1

4.1 Introduction

3D human shape and cloth modeling is an active area of research with wide applications in AR/VR domain. Efforts like [27, 25, 72, 47, 46, 71, 74, 32, 21, 64, 65] reconstruct realistic clothed humans from images as garments play an important role in inducing realism in digital avatars. However, the majority of existing works assume the availability of synthetic parametric garment meshes [66, 3, 60]. Some of the nascent efforts on garment digitization [73, 58, 31, 11, 14, 41, 40, 5] focus on extracting high-fidelity 3D garments from monocular images. Furthermore, the next key challenge is to learn automated retargeting or draping of garments over digital avatars, which has use-cases in fashion design, animation, gaming, and, most importantly, virtual try-on (VTON). Unlike image-based 2D VTON solutions [13, 33], 3D garment retargeting offers a more controllable and elegant solution leading to an immersive experience for AR/VR environments.

3D garment retargeting aims at realistically draping a 3D garment over a target human body in varying shapes and poses by inducing geometrical deformations over the garment surface arising due to such changes. This problem is challenging because of several factors: arbitrary body shapes and poses, modeling topological differences among various categories of garments along with realistic deformations arising out of the physical interaction with the underlying body and resolving the penetration/intersection of the garment with the underlying body.

Existing solutions for 3D garment retargeting can be largely classified into traditional graphics simulation pipelines [7, 57, 56] and modern deep learning techniques [9, 6, 8, 10]. While the simulationbased approaches provide an accurate detailing of deformation and wrinkles, they often rely on the previous frames of reference to obtain velocity and acceleration parameters. However, a fashion shopper on AR/VR application would desire to see how a garment would look when draped onto their digital avatar. Thus, asking the user to provide a trajectory for an accurate drape might not be feasible solution and hence requiring an arbitrary pose and shape re-targeting technique. Some of the recent deep learning-based efforts like [10] have made progress in this direction utilizing supervised training strategies learning the skinning weights of the parametric garment for draping it onto a parametric human body. They consider SMPL [38] as the parametric body model, and garments are also derived from the SMPL body mesh [43, 37, 16]. However, there are several issues with such approaches. First, a large number of training samples are required, generated by simulating garments on top of body mesh using software, e.g. Blender. The parametric garments are synthetic in nature and fail to model garments with arbitrary topologies. Additionally, a garment and body extracted from a real scan or reconstructed from an image (using [73, 58, 5]) are not parametric in nature and the aforementioned approaches cannot handle them. HOOD is a parallel work that doesn't handle non-parametric cases, hence not applicable to real-world settings

In this work, we propose a self-supervised method that can retarget real, non-parametric garment meshes over a target human body (either parametric or non-parametric). Given a 3D garment mesh and a target 3D human mesh, we first estimate correspondences between the two meshes using a novel representation, which provides an initial placement of the garment around the target body as a coarse



Figure 4.2: Outline of the proposed garment retargeting method.

retargeting initialization. We then employ a self-supervised training strategy, where we refine the coarse initialization and model shape and pose-specific deformations by minimizing the standard physics-based losses. Unlike existing methods, our framework doesn't learn skinning weights, therefore, can repose any arbitrary non-parametric garment on any parametric or non-parametric target body. Finally, as a post-processing step, we explore two different approaches for introducing fine wrinkles in the retargeted garment and provide qualitative comparison between the two. Additionally, due to the lack of any real-world datasets for 3D garment retargeting, we curate our own dataset captured using a multiview Azure Kinect RGBD setup, containing different garments worn by multiple subjects in arbitrary poses. Our dataset serves as the ground truth for evaluating the proposed method for 3D garment retargeting. In summary, our main contributions are:

- We present a novel framework for retargeting real, non-parametric 3D garments on an arbitrary target human body.
- Our novel formulation can retarget in-the-wild, non-canonical and non-parametric garments over any arbitrary target body.
- We propose a first-of-its-kind real-world 3D garment dataset for evaluating our proposed framework.

We plan to release both the dataset and the code to further democratize research in this domain.

4.2 Related Work

Several 2D VTON methods exist [24, 61, 67, 54], which employ deep generative learning for draping 2D garments over 2D human images. Specifically, such approaches first aim to segment garments from an input image space, performing an initial thin-plate-spline (TPS) based transformation to roughly warp and align the garment onto the target image. Then, the transformed garment image is blended with the target person's body image to generate a realistic try-on image, generally using image-to-image translation networks. Generative networks tend to produce blurry results and artifacts; even when high-resolution modeling [33, 13] is employed. Moreover, 2D VTON methods have limited ability in terms of adjusting the pose and viewpoint for a more immersive experience. A recently proposed work StylePose[2] has the ability to *repose* the clothed humans to a novel viewpoint in image space leveraging partial 3D priors. However, the work does not allow accurate and view-consistent draping of the 2D garments over a different person altogether, thereby not meeting the basic requirement of a VTON solution. Moreover, to our knowledge, any 2D VTON solution would fail to preserve the accurate geometry of the garment after the transformation.

Clearly, the exploration of 3D space is a more viable option to tackle these challenges. 3D-VTON solutions offer the ability to preserve the geometry of the garments and easily allow change of garment, and pose properties and viewpoints. However, there is a significant white space in the area of 3D-VTON research. 3D VTON can be seen as transforming a garment in 3D Euclidean space, in order to align it over (or around) a target 3D human body (SMPL mesh, 3D scan etc.), while avoiding intersections of the garment with the target body. It is highly desirable to model deformations in the garment corresponding to the target body's pose and shape. Recently, state-of-the-art works like [69] claim to propose the first 3D VTON solution by extending the 2D TPS-driven generative pipeline to reconstruct the 3D geometry, finally blending on a try-on image, with a representation similar to that of Moulding-Human [18]. Although this allows viewing the draped garment on the target body from arbitrary viewpoints, the draping is still performed in image space using GANs and hence suffers from limitations such as blurry artifacts and false geometrical deformations. Additionally, since the method starts from the image of a garment, extending it to a real-world scan of a 3D garment is not trivial.

On the other hand, several deep learning methods [50, 19] have been proposed, which learn to simulate a 3D garment mesh onto a 3D body mesh, as the classical mass-spring model which is computationally expensive and slow. [50] and [9] rely on self-supervised physics-based losses in order to model the dynamics of a garment due to changes in the underlying body pose. More specifically, these methods learn to transform the 3D garment mesh over a gradually animating 3D body (a smooth pose-change trajectory). Such a formulation, replicating cloth simulation, will not be able to deal with sudden changes in the pose and shape.

Methods like TailorNet[43] have made promising progress towards 3D instance try-ons and followup works [14, 16, 37] have extended incorporating a corpus of garments as a latent representation. However, all of them rely on synthetic or parametric garments rather than dealing with real-world scanned garments. While it is true that extending these works to real-world garments is challenging, valida-



Figure 4.3: Results from our method for retargeting 3D garment onto SMPL body meshes of different poses and shapes (a) - (f); and on non-parametric 3D human scans (g) (h).

tion of the leveraged technique is also a significant challenge. As most commonly available multi-pose clothed-human datasets either provide synthetic and parametric clothing[4, 44] or lack garment-specific shape variation[39, 15].

4.3 Method

Figure 4.2 outlines our proposed framework, which has three key modules, namely, Correspondenceguided coarse retargeting, Self-supervised refined retargeting, and Wrinkle generation. The input garment and the target body are fed to the first module, which exploits an intrinsic mesh characterization to estimate dense correspondences between them, providing an initial coarse retargeting. This coarse retargeted garment is subsequently passed to our self-supervised refinement network, which refines the garment mesh geometry and introduces target body-specific surface deformations. Finally, the refined, retargeted garment is further processed with an optional wrinkle generation module to generate pose and shape-specific wrinkles if required.

4.3.1 Correspondence-Guided Coarse Retargeting

The aim of this module is to perform a coarse retargeting of the garment mesh over the target body mesh by first establishing dense surface-level correspondences between the two. Utilizing these correspondences, we transform the garment mesh vertices to align with the target body mesh vertices. The key idea is to establish dense correspondences which can provide a *coarse* understanding of how the garment should be draped on the target body; e.g., sleeves going around the arms, the collar going around the neck etc. SMPL[38], being a parametric body model, is a natural choice for acting as a medium for establishing dense surface correspondences, as it can easily model variations in human shapes and poses. Therefore, we first perform dense non-rigid registration of both garment and target body mesh with the SMPL mesh. More details on SMPL registration is provided in subsection 4.4.1.

Let the garment mesh be \mathcal{G} , target body mesh be \mathcal{T} and their corresponding SMPL meshes be $\mathcal{M}_{\mathcal{G}}$ and $\mathcal{M}_{\mathcal{T}}$, respectively. Establishing correspondences between \mathcal{G} and \mathcal{T} simply means for each vertex $v_i \in \mathbb{R}^3$ of \mathcal{G} , locating a 3D point $x_i \in \mathbb{R}^3$ on the surface of \mathcal{T} , where v_i should be coarsely placed.

To achieve this we first define global features ϕ_i for each vertex q_i of the SMPL meshes $\mathcal{M}_{\mathcal{G}}$ and $\mathcal{M}_{\mathcal{T}}$, extrapolate these features to the vertices of \mathcal{G} and \mathcal{T} , and then perform correspondence matching based on these features. More specifically, the task is to estimate a feature vector $\phi_{smpl} = [\phi_1, \phi_2, ..., \phi_{6890}]$ for each vertex q_i of SMPL mesh, where $\phi_i \in \mathbb{R}^d$. ϕ_{smpl} is same for any SMPL mesh registered with any garment or body, i.e. $\phi_{smpl} = \phi_{\mathcal{M}_{\mathcal{G}}} = \phi_{\mathcal{M}_{\mathcal{T}}}$. Then, feature vector for each vertex v_i of \mathcal{G} is computed as follows:

$$\phi_{\mathcal{G}}^{i} = \frac{\sum_{j=1}^{k} [\phi_{\mathcal{M}_{\mathcal{G}}}^{j} / dist(v_{i}, q_{j})]}{\sum_{j=1}^{k} [1 / dist(v_{i}, q_{j})]}; q_{j} \in \mathcal{N}^{i}$$

$$(4.1)$$

$$\mathcal{N}^{i} = [q_{1}, q_{2}, ..., q_{k}] \tag{4.2}$$

where, dist() is the \mathbb{L}_2 distance, q_j is a vertex of $\phi_{\mathcal{M}_{\mathcal{G}}}$ and j^{th} nearest neighbor of v_i in Euclidean space; and $|\mathcal{N}^i| = k = 32$ (set empirically). Similarly, we compute $\phi_{\mathcal{T}}$ by extrapolating $\phi_{\mathcal{M}_{\mathcal{T}}}$ based on k-nearest neighbor distance.

Few essential aspects to be taken into consideration for choosing appropriate ϕ_{smpl} are as follows, First, the feature embedding ϕ_{smpl} should incorporate both the local neighborhood information, while maintaining global structural context. Moreover, it should be concise yet representation-rich to uniquely characterize the associated surface, especially when extrapolating to the registered garment mesh or target body mesh. Additionally, ϕ_{smpl} should be continuous over the surface of SMPL mesh to ensure locally smooth encoding of neighborhood information. We experimented with existing representations such as CSE[42] and BodyMap[28] to serve the need for ϕ_{smpl} , as they promise to encode global structural information. However, we empirically found them to produce false matching due to the repetition of extrapolated features (due to very low dimensionality). A detailed study on this is provided Table 4.6.2.

Therefore, we develop a new strategy to establish correspondence across different garments and human body via SMPL, leveraging the intrinsic geometry-based Isomap Embeddings[30]. In order to encode local neighborhood information, we first compute the pairwise geodesic distance matrix, $|\mathbb{D}_{geo}| = 6890 \times 6890$, for all pairs of vertices (q_i, q_j) of the SMPL mesh; i.e.

$$\mathbb{D}_{geo}{}^{ij} = geodist(q_i, q_j) \tag{4.3}$$

To incorporate global information, we use isometric mapping to fit the vertices of SMPL mesh onto a *d* dimensional manifold by extending metric multi-dimensional scaling (MDS) based on \mathbb{D}_{geo} . This gives us a *d*-dimensional representation of each SMPL vertex q_i , i.e. ϕ_{smpl} . We empirically found that setting *d*=128 ensures sufficient dimensionality to avoid repetitions while extrapolating on the target or registered mesh. Finally, we estimate $\phi_{\mathcal{G}}$ and $\phi_{\mathcal{T}}$ using Equation 4.1. These extrapolated features are termed as *Isomap Embeddings*.

Based on the estimated *Isomap embeddings*, we first perform an initial retargeting to *coarsely* place the garment around the target body. In particular, for each vertex v_i of \mathcal{G} , the corresponding 3D target location x_i in the vicinity of \mathcal{T} is estimated as follows:

$$x_i = \frac{\sum_{j=1}^k [u_j/dist(\phi_{\mathcal{G}}^i, \phi_{\mathcal{T}}^j)]}{\sum_{j=1}^k [1/dist(\phi_{\mathcal{G}}^i, \phi_{\mathcal{T}}^j)]}; \phi_{\mathcal{T}}^j \in \mathcal{N}^i$$
(4.4)

$$\mathcal{N}^{i} = [\phi_{\mathcal{T}}^{1}, \phi_{\mathcal{T}}^{2}, ..., \phi_{\mathcal{T}}^{k}]; \phi_{\mathcal{T}}^{j} \in \phi_{\mathcal{T}}$$

$$(4.5)$$

where, u_j is $j^t h$ vertex of the target body, dist() is the L2 distance, \mathcal{N}^i the set of k-nearest neighbors of $\phi_{\mathcal{G}}^i$ in $\phi_{\mathcal{T}}$, and $|\mathcal{N}^i| = k = 32$. We replace the vertices v_i of $\phi_{\mathcal{G}}$ with corresponding x_i , coarsely retargeting the garment mesh around the target mesh $\phi_{\mathcal{T}}$. This coarse initialization is then refined using a self-supervised strategy explained in the next section.

4.3.2 Self-Supervised Refined Retargeting

Given a coarsely retargeted garment mesh, where the garment vertex mesh coordinates v_i are replaced by their respective correspondence surface points x_i on target body mesh, we propose to refine these vertex positions further to incorporate accurate pose and shape-specific deformations, and also to restore the structural geometry of the garment. However, supervised learning is not suitable for this refinement task due to the lack of ground truth pairs on real data. Thus, we resort to a self-supervised setup where we minimize losses that retain the structural integrity of the garment mesh (namely, retaining edge lengths and relative face orientation) while preserving the coarse retargeting.

Let the refined vertex positions of the garment mesh \mathcal{G}' be $v'_i = x_i + \Delta x_i$. We employ a Multi-Layer Perceptron (MLP) network to predict per-vertex $\Delta x_i \in \mathbb{R}^3$. The per-vertex input to the MLP is $\mathcal{I} = \{x_i, \phi^i_{\mathcal{G}}, \chi^{k,i}_{\mathcal{M}_{\mathcal{T}}}, \psi_{\mathcal{G}}, \psi_{\mathcal{T}}\}$. Here, $x_i \in \mathbb{R}^3$ is i^{th} vertex-position of the coarsely retargeted mesh and $\phi^i_{\mathcal{G}} \in \mathbb{R}^{128}$ is the corresponding isomap embedding. Additionally, the MLP also takes k-nearest neighbours of x_i belonging to the vertex set of target body mesh \mathcal{T} , denoted as $\chi^{k,i}_{\mathcal{M}_{\mathcal{T}}}$ (k = 32). In order to encode a useful global context for both garment and target body, we use two separate PointNet[45] encoders, which provide 128 dimensional global encoding of the vertices of the garment mesh and the body mesh, denoted as $\psi_{\mathcal{G}} = PointNet_{\mathcal{G}}(vertices(\mathcal{G}))$ and $\psi_{\mathcal{T}} = PointNet_{\mathcal{T}}(vertices(\mathcal{T}))$, respectively. Both the encoders are trained jointly with the MLP decoder in a self-supervised fashion to minimize the following losses:



Figure 4.4: Results of real garments draped on unseen pose and shape.

Edge-Length loss: This loss is used to preserve the structural integrity of the garment by constraining the change in the length of the edges of the original garment mesh, calculated as follows:

$$\mathcal{L}_{length} = \frac{1}{m} \sum_{i=1}^{m} w_i \cdot \left\| e_i - e'_i \right\|$$
(4.6)

$$w_i = \begin{cases} 0 & \text{if } e_i \in \mathbf{J} \\ 1 & \text{otherwise} \end{cases}$$
(4.7)

where, $e_i \in edges(\mathcal{G})$, $e'_i \in edges(\mathcal{G}')$ and $m = |edges(\mathcal{G})|$. J is the set of edges of the garment mesh belonging to the special joint locations of the underlying human body, specifically, elbows, armpits, waist, and knees as shown in Figure 4.16. These are the prominent regions that undergo extreme deformation due to pose change. Hence, we chose not to preserve edge length around such regions to allow accurate reposing of the garment.

Correspondence Loss: Edge-length loss has the effect of retaining the original pose and shape of the garment in order to maintain its structure. We employ an additional loss to constrain this behavior by ensuring that the correspondences between the refined garment and the target body should be similar as for the original garment used for coarse retargeting. The predicted residual Δx_i is used to get refined vertex positions $v'_i \in \mathcal{G}$. We then compute correspondences x'_i for each v'_i using Eq. Equation 4.4 and minimize the L2 norm between x_i and x'_i , i.e.

$$\mathcal{L}_{corres} = \frac{1}{n} \sum_{i=1}^{n} \left\| x_i - x'_i \right\|; \ n = |vertices(\mathcal{G})|$$
(4.8)

It ensures that the garment doesn't deviate too much away from the initial coarse retargeting and remains in the vicinity of the target body.

Bend Loss: We impose bend loss, introduced in [50], to ensure that the angle between two adjacent faces is as low as possible. This makes sure that the output is smooth and does not have any weird deformations or artifacts.

4.3.3 Wrinkle Generation Module

We propose the following two methods for wrinkle generation and provide a detail study of the effects of these on retargeted garments.

4.3.3.1 Supervised Wrinkle Generator

The retargeted refined garment obtained from the previous module has a smooth surface and lacks realistic wrinkles, which are an inherent part of garment geometry. Thus, as an optional post-processing step, we learn to induce plausible wrinkles on the garment mesh, conditioned on the target body pose and shape. We propose a supervised MLP for generating/inducing pose and shape-specific wrinkles on the



Figure 4.6: Inference on T-shirt.

Figure 4.5: Wrinkle generation module architecture.

garment surface by learning from real-world data. We choose DiffusionNet[52] encoder for encoding garments due to its ability to capture both local high-frequency details as well as the global context by simulating the diffusion process at multiple scales. We also use PointNet[45] encoder to encode the vertices of the target body mesh. These 128-dimensional embeddings act as a conditioning prior so that the wrinkles should be generated based on the geometry of the target body.

Given the refined retargeted garment \mathcal{G}' draped over the target body \mathcal{T} , the task is to estimate detailed garment \mathcal{G}'' , which is essentially adding residual to the vertices of \mathcal{G}' , i.e. $v''_i = v'_i + \delta_i$, where $\delta_i \in \mathbb{R}^3$. For each vertex v'_i of \mathcal{G}' , we estimate per-vertex DiffusionNet encoding $\psi_{\mathcal{G}'_i}$ and for target body mesh \mathcal{T} , we estimate global encoding $\psi_{\mathcal{T}}$ using PointNet. The MLP-decoder takes v_i , $\psi_{\mathcal{G}'_i}$, and $\psi_{\mathcal{T}}$ as input and predicts δ_i .

We train jointly train PointNet, DiffusionNet and MLP-decoder networks in a supervised manner on THuman2.0 and 3DHumans datasets. As a part of data preparation, we first manually segment and extract garments \mathcal{G}_{GT} from the 3D scans. Then, we perform Laplacian smoothening[17] on the garments to get rid of wrinkles and other details. During training, the smoothened-out garment \mathcal{G}_{smooth} and its underlying SMPL mesh $\mathcal{M}_{\mathcal{G}}$ are passed to the network to predict per-vertex residual. The residuals are added to vertices of \mathcal{G}_{smooth} to obtain $\mathcal{G}_{detailed}$. The following losses are used during the training:

L1 Loss: L1 loss minimizes the L1 norm between the vertices of $\mathcal{G}_{detailed}$ and \mathcal{G}_{GT} .

Normal loss: Normal loss maximizes the cosine similarity between the vertex normals of $\mathcal{G}_{detailed}$ and \mathcal{G}_{GT} . In order to focus the network on areas with high extrinsic curvature, we mask out the loss for vertices belonging to locally flat regions, based on the normals of their neighborhood vertices.

Laplacian loss: We propose a novel Laplacian loss to capture high-frequency details of the garment. First, we compute the residual in Laplacian coordinates as:

$$v_i^{laplace} = v_i'' - \frac{1}{|\mathcal{N}_i|} \sum_{j=1}^{|\mathcal{N}_i|} v_j''$$
(4.9)

$$\delta_i^{laplace} = ||v_i'' - v_i^{laplace}|| \tag{4.10}$$

where j is the neighborhood per vertex, ||.|| is euclidean between the original mesh and Laplacian smoothed mesh. We take the L2 norm between the delta of the ground truth and the predicted calculated as above. This way, we impose that the network captures the high-frequency details. Once training is completed, for inference, \mathcal{G}' is passed to the wrinkle generation module to obtain \mathcal{G}'' . Result is shown in Figure 4.6

4.3.3.2 Laplacian Detail Transfer

We observe that our learning-based wrinkle generator network doesn't preserve garment details like pockets, collar etc. It also needs to be trained in supervised fashion. Hence we propose *Detailed Preservation Module* that adapt [55], to preserve the high-fidelity geometric details of the input garment and integrate it with the refined retargeted garment. Given the input garment mesh \mathcal{G} with $V_{\mathcal{G}} = \{v_1, v_2, ..., v_N\}$ vertices in \mathbb{R}^3 where \mathcal{N} is the total number of vertices the Laplacian Matrix can be used to retrieve the high fidelity details of the mesh. For each vertex v_i let, $\mathcal{N}_i = \{j | (i, j) \in K\}$ be the neighborhood ring directly connected to v_i and degree d_i be the number of vertices in \mathcal{N}_i . The uniform Laplacian coordinate per vertex is given as:

$$\delta_i(v_i) = v_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_k} v_j \tag{4.11}$$

The above equation can be represented in matrix form: $L[v_1, v_2, ..., v_N]^T = [\delta_1, \delta_2, ..., \delta_N]^T$ where L is the uniform Laplacian Matrix given as $L = I - D^{-1}A$. Here A is the mesh adjacency matrix and $D = diag(d_1, d_2...d_N)$ be the degree matrix.

In order to integrate the high-fidelity geometric details from input garment on to retargeted garment, we first calculate the uniform Laplacian Matrix $L_{\mathcal{G}}$ and Laplacian coordinates $\delta_{\mathcal{G}}$ of the input mesh \mathcal{G} . We fix anchor points on the retargeted mesh \mathcal{G}' and recompute the Laplacian matrix as $\hat{L} = [L_{\mathcal{G}}^T, 1_i]^T$ and Laplacian coordinates as $\hat{=} [\delta_{\mathcal{G}}, v_i]^T$. 1_i is the one hot encoding where i_{th} is one. We finally obtain the retargeted mesh with high fidelity details \mathcal{G}'' with $V_{\mathcal{G}''}$ vertices by solving a linear system to obtain the modified vertex positions as $V_{\mathcal{G}''} = \hat{L}^{-1}\hat{\delta}$. We show the result of Detail Preservation module in Figure 4.8

We provide a comparison between the two methods in Figure 4.9. We observe that Laplacian Detail Transfer method better preserves the input details. Hence we use this as post-processing module to get wrinkles in all the following results.



Figure 4.7: The figure shows different parametric cloth setting both tops and bottoms draped onto SMPL extracted from the AMAAS Dataset.



Figure 4.8: Results of Laplace detail integration.

4.4 Implementation Details

4.4.1 SMPL Registration:

In order to establish the dense correspondences for coarse retargeting of the mesh, we first estimate the pose & shape of the underlying body in both meshes (the *garment* as well as the *target* body). If the garment or the target body is already present in canonical pose and shape, then the SMPL parameters can be directly picked from the canonicalized SMPL. In the absence of canonicalized meshes (garments or target bodies), we employ a similar SMPL fitting strategy as proposed by PAMIR[71] for obtaining SMPL body parameters. The pipeline of PAMIR extends the SMPL fitting methodology of [59] exploiting multi-view consistency. The resultants are registered SMPL bodies for both the garment and target-body meshes. *It is to be noted that, despite massive efforts to employ multi-view consistency, the registration pipeline is far from accurate*. Our framework is robust enough to handle noise in pose & shape parameters. Finally, the estimated pose & shape parameters are used to generate SMPL mesh \mathcal{M} , consisting of 6, 890 vertices and 13, 776 faces. This step is important for estimating isomap embeddings for each vertex of the garment using k-nearest-neighbor extrapolation of SMPL vertices. The process is shown in Figure 4.11



Figure 4.9: Comparison between two proposed wrinkle generation modules.

4.4.2 Refined Retargeting Module

The coarse retargeted mesh obtained using dense correspondence between garment and target body is refined using a self-supervised *Refined Retargeting Module*. It is composed of two PointNet encoders $PointNet_{\mathcal{G}}$ and $PointNet_{\mathcal{T}}$ for encoding both input garment and target body respectively and an MLP decoder. The PointNet encoder consists of 5 ResNet blocks with skip connections between each block. Each ResNet block is an FC (fully connected) layer with ReLu activations. Each encoder outputs a latent code of 128-dimension. These encodings along with the coarsely initialized garment vertices, *k*-neighbours of target mesh, and the iso-embedding of the input garment are fed to the MLP decoder. The MLP is constituted of six hidden layers with 512 neurons each activated by LeakyReLu functions. The last layer of MLP is a Tanh.

Apart from feeding PointNet features of the garment and body as input, we also condition every layer of the MLP with PointNet features similar to ADAIN[26]. The MLP outputs a Δx value which is added to the *course-retargeted* mesh to obtain *refined-retargeted* mesh.

4.4.3 Supervised Wrinkle Generation Module

As an optional post-processing step, we learn to induce plausible wrinkles on the garment mesh, conditioned on the target body pose and shape. The smooth input garment is encoded using DiffusionNet



Figure 4.10: Retargetting 3D garments from CLOTH3D dataset onto non-parametric human bodies from THumans2.0[68] dataset. Our approach can deal with layered clothing as well.



Figure 4.11: This shows the propogation of Isoembeddings from intersection points to the whole body.



Figure 4.12: The figure shows different real scanned garments of our *Dress Me Up* dataset draped onto SMPLs of AMAAS dataset

[52] which is effective in learning high-frequency details. The DiffusionNet consists of 4 blocks, with a channel width of 128 and 128 eigenbasis vectors for spectral acceleration. We use ReLU activations at intermediate layers and softmax for the final layer. We also use PointNet encoders, similar to one used in *Refine Retargeting Module* for encoding target vertices. We use a decoder MLP with six hidden layers of 512 neurons each. The MLP is activated by LeakyReLu activation in the internal layers and with a Tanh final layer. The MLP takes the DiffusionNet and PointNet encodings as input and outputs a δ value per vertex which is added to the input-smooth garment to obtain plausible wrinkles. Refer Figure 4.5

4.4.4 Datasets

To evaluate our approach, we require ground truth 3D garments to be draped over the target body of poses and shape variations. However, as mentioned earlier, there is a significant lack of such large-scale

datasets. CLOTH3D is the only dataset that offers data in the required setting. However, the garments are synthetic and parametric in nature, draped using a simulated engine. Hence the lack of real-world aesthetics and noise is prevalent. To address this gap, we capture our own dataset "DressMeUp" to validate our approach on a real-world data distribution. We briefly describe both datasets, and additional details are provided in subsection 4.6.1

CLOTH3D: Cloth3D provides a simulated collection of sequences containing clothed humans, modeled using SMPL meshes and their corresponding parametric garments. They model the animations in accordance to a large collection of MoCap data. The dataset offers a wide garment range(t-shirts, tank-tops, trousers etc.) which we broadly group into two categories – TopWear and BottomWear.

DressMeUp (Our Dataset): As stated earlier in section 4.1, there is a need for real-world 3D garment datasets to validate the proposed methodologies, which contain realistic garments draped on real humans. To bridge this gap, we captured around ~ 255 meshes of real garments draped onto humans of varied poses and body profiles. We show some sample meshes of this dataset in Figure 4.19 and Figure 4.20. We believe that this dataset provides a more rigorous evaluation, extending beyond the parametric modeling of clothing and latent garments.

This data was captured using Azure Kinect-based multiview RGBD capture setup. We collected ~ 255 garments scans, worn by 15 unique subjects, with 44 unique garments. For every garment, a subject is scanned in 5 different poses. Each pose is captured using a static multi-view(7) RGBD system. To obtain final mesh reconstructions, we employ multiview Kinect Fusion[29] on the captured RGBD data. To further rectify the noise of the raw scan, manual post-processing is performed utilizing the eclectic and elegant toolkit of Meshlab. While post-processing, we also obtain a UV-mapped mesh of the garment to facilitate texture swapping. Additionally, we perform SMPL registration for each mesh to approximate the pose and shape. Our dataset captures realistic noise and topological deformations of real-world garments draped over different subjects under different poses. We believe our dataset can prove to be extremely useful in the progress of the 3D-VTON domain.

4.4.5 Evaluation Metrics

To quantitatively evaluate our proposed approach, we report widely used metrics like Euclidean Distance(ED), Normal Consistency(NC), Interpenetration Ratio(IR) and Point-to-Surface Distance(P2S).

Given a 3D garment mesh \mathcal{G} to be retargeted and the corresponding GT garment mesh \mathcal{G}_{GT} (where $v_i \in vertices(\mathcal{G})$ and $\hat{v}_i \in vertices(\mathcal{G}_{GT})$), we use the following standard metrics for evaluation: **Euclidean Distance(ED):** We compute ED as the average Euclidean distance between the correspond-

ing vertices of input and final retargeted garment mesh, i.e.

$$ED = \frac{1}{n} \sum_{i=1}^{n} \|v_i - \hat{v}_i\|$$
(4.12)

Lower values for ED are desired for better output.

CLOTH3D					
Τυρε	P2S↓	ED↓	NC↑	IR%↓	
x 10 ⁻³					
topwear	6.901	9.353	0.951	0.009	
bottomwear	8.049	9.832	0.943	0.006	
OUR CAPTURED DATA					
topwear	12.119	12.571	0.854	0.037	
bottomwear	6.753	7.314	0.849	0.014	

Table 4.1: Quantitative evaluation of topwear-TOPWEAR and bottomwear-BOTTOMWEAR both in the case of Cloth3D and our Dress-Me-Up data.

Normal Consistency(**NC**): We compute NC as the average cosine similarity between the corresponding vertex normals of input and final retargeted garment mesh, i.e.

$$NC = \frac{1}{n} \sum_{i=1}^{n} n_i \cdot \hat{n}_i$$
 (4.13)

Values close to 1 are desirable for NC.

Interpenetration Ratio(**IR**): It is computed as the ratio of the area of garment faces inside the body to the overall area of the garment faces, hence lower values are desired to ensure the least amount of penetration of the garment mesh with the target body mesh.

Chamfer Distance (CD): Given two sets of points S_1 and S_2 , Chamfer distance measures the discrepancy between them as follows:

$$CD = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$
(4.14)

In our case, $S_1 = vertices(\mathcal{G})$ and $S_1 = vertices(\mathcal{G}_{GT})$.

Point-to-Surface (P2S) Distance: P2S measures the average L2 distance between each vertex of the garment mesh and the nearest point to it on the target body surface.

4.5 **Experimentation and Results**

4.5.0.1 Qualitative and Quantitative Results on CLOTH3D:

For evaluation purposes, we randomly select \sim 273 random sequences from the CLOTH3D dataset. We uniformly sample 5 frames per sequence, ensuring that there is a significant *pose* change among



Figure 4.13: Comparison of our method with M3DVTON[69] for draping non-parametric garments. M3DVTON introduces false garment geometry (the sleeve of the t-shirt mapped to the sleeveless part of the target geometry) to inaccurate geometries.

the sampled frames. Out of five sampled frames, we take SMPL bodies from the first three for selfsupervised training and use the remaining for evaluation. Additionally, instead of taking garments from each sequence, we *only* sample 10 garments out of the available corpus of garments for self-supervised training, to ensure evaluation is only done on unseen garments. Figure 4.3 shows qualitative results of our framework on CLOTH3D dataset, where we report retargeting results on three different poses along with three different shapes. In Figure 4.7, we show results of garments draped onto three distinctive and challenging SMPL poses obtained from AMAAS[39] dataset. Do note that we also demonstrate our results of bottom wear. Our framework can retarget arbitrary unseen garments on the target bodies with varying poses and shapes, as evident from the figures. We also report quantitative metrics mentioned in subsection 4.4.5 on the evaluation samples of CLOTH3D in Table 4.1. We achieve sufficiently low ED, P2S, and IR metrics while maintaining high Normal Consistency.

4.5.0.2 Quantitative Results on Our Dataset:

For evaluation of our dataset, we perform self-supervised training on 500 target SMPL meshes from AMASS dataset to ensure enough pose variation, minimizing losses while learning to drape 10 synthetic garments from CLOTH3D dataset. Even being trained on synthetic garments, our network is able to generalize on real garments from our dataset. Table 4.1 reports corresponding evaluation metrics where we achieve satisfactory performance. The values reported on CLOTH3D are slightly better because training and evaluation are both done on synthetic garments. However, in the case of our dataset, training is done on synthetic garments and evaluation on real garments, thereby leaving a window for an out-of-distribution scenario.

4.5.0.3 Qualitative Results on Real Scans:

Figure 4.4 shows qualitative results of our framework on real garments retargeted to arbitrary SMPL meshes, and Figure 4.1, Figure 4.15 shows qualitative results on real target human scans. We also show results of our method on real scans of the THumans2.0 dataset in Figure 4.10. In Figure 4.12, we show our real-world scan being draped onto SMPLs of AMAAS data. We show the results of DressMeUp garments draped on real scans of THuman2.0 dataset in Figure 4.21. It is evident from both the figures that even being trained on synthetic garments and target SMPL meshes, our framework can retarget real garments on arbitrary real scans (not just SMPL meshes). This highlights the generalization capabilities of our framework on real-world samples. We can also drape garments on top of other garments, hence making way for layered clothing as well.

Qualitative Results on Internet Images: We additionally show results on garments extracted from internet images to highlight the application of our framework towards image-based 3D VTON methods (e.g. M3DVTON[69]). We use the recently proposed method [73] to extract 3D garments, and [64] to reconstruct 3D humans from images. Figure 4.14 shows qualitative results of retargeting 3D garments



Figure 4.14: Qualitative results of our garment retargeting method on non-parametric avatars reconstructed from internet images.



Figure 4.15: Comparison with M3DVTON.

Figure 4.16: Joint Masks

on 3D Humans both reconstructed from images. Note that we only show geometry as both [73] and [64] don't retain texture information. This is yet another proof of good generalization of our method.

Table 4.2:	Noise	ablation	in o	corresponder	nce
estimation					

Naisa	P2S↓	ED↓	NC↑	IR%↓
noise	x 10	-3		
10^{-4}	7.481	9.544	0.934	0.009
10^{-3}	7.521	9.581	0.927	0.009
10^{-2}	10.247	11.97	0.761	0.014

Table 4.3: Ablation on number of input trainingsamples.

Locatura	P2S↓	ED↓	NC↑	IR%↓
Loss type	$x \ 10^{-3}$			
10 garments	6.901	9.353	0.951	0.009
50 garments	7.370	9.511	0.934	0.008

4.5.1 Comparison

M3DVTON: Figure 4.15 shows a comparison of M3DVTON[70] with our framework on random internet images (as mentioned earlier, we use off-the-shelf methods to extract 3D garments and target human body). It is evident from the figure that since M3DVTON performs retargeting in 2D space,



Figure 4.18: (a) Jump and (b) Smooth trajectory.

it doesn't produce accurate geometric deformations. Moreover, since it uses a supervised keypoint detection method for initial TPS-based draping, it suffers when the target subject's garment category doesn't match the source garment category. However, our method doesn't suffer from such limitations and can retarget arbitrary garments on arbitrary targets. We show additional results in comparison with the M3DVTON method. Some more examples are shown in Figure 4.13

Draping Implicit Garments(DIG): Figure 4.17 shows qualitative comparison of our method with DIG[37]. As the training code for DIG, or any other Implicit method, e.g., [16] is not available, we couldn't compare quantitatively. However, qualitatively our results are on par if not superior, to DIG as evident in the figure.

Neural Cloth Simulation: As discussed in section 4.2, simulation-based approaches require a smooth continuous trajectory to repose the garments from one pose to another. When it comes to going from one extreme pose to another (jump trajectory), such methods fail to generate accurate deformations and wrinkles, while also causing severe interpenetrations. We show this effect in Neural Cloth Simulation [9] in Figure 4.18. Moreover, they offer no provision for draping a garment on an entirely new subject, hence no compliance for 3DVTON solutions. Whereas, our framework can retarget any garment from one extreme pose to another extreme pose, even on an entirely new subject.

Lass truck	P2S↓	ED↓	NC↑	IR%↓
Loss type	x 1() ⁻³		
\mathcal{L}_{corres} only	7.406	9.593	0.935	0.0217
\mathcal{L}_{length}	9.614	11.352	0.932	0.058
\mathcal{L}_{bend} only	10.245	11.923	0.928	0.104
Without \mathcal{L}_{corres}	12.125	13.445	0.929	0.135
Without \mathcal{L}_{length}	10.560	11.940	0.933	0.022
Without \mathcal{L}_{bend}	7.406	9.593	0.935	0.021
Without Joint Mask	10.560	11.941	0.933	0.022

Table 4.4: Effect of different losses.

4.5.2 Ablation Studies

In this section, we discuss the ablation on self-supervised losses of the refinement module and an ablative study of the supervised wrinkle generation module.

Noise in Correspondence Estimation: We analyze the effect of noise in correspondence estimation by introducing noise at different levels. For each correspondence pair (v_i, x_i) we add Gaussian noise to x_i with zero mean and varying standard deviation, i.e. $x_i = x_i + \mathcal{N}(0, \sigma)$; $\sigma = \{0.001, 0.01, 0.1\}$. Please note that for brevity we are writing the 3D noise vector as $\mathcal{N}(0, \sigma)$ since $x_i \in \mathbb{R}^3$. We then pass the noisy coarse initialization to the further modules and compute the evaluation metrics (combined for topwear and bottomwear), reported in Table 4.2. As can be seen, our framework is robust enough to handle noise with $\sigma = 0.001, 0.01$, where the evaluation metrics are on par with the noise-free setting. However, with $\sigma = 0.1$, the performance of the method drops.

4.5.2.1 Ablation on Self-Supervised Losses

We provide an ablative study of the effect of each loss and report the relevant metrics in the Table 4.4.

4.5.2.2 Ablation on Supervised Wrinkle Generation Module

We provide a quantitative evaluation of the performance of our wrinkle generation network on two datasets 3DHumans [15] and THuman2.0 [68]. We divide both the datasets into train and test splits and report the P2S distance, Euclidean distance and Normal Consistency losses. Refer to Table 4.5

4.6 Discussion

4.6.1 Description of DressMeUp Dataset

We provide our own textured garment dataset, curated using Kinect cameras. The dataset consists of 50 different garments, with 44 unique garments worn by 15 individuals. Each garment is provided in 5 different poses on the same person, resulting in a total of 250 garment meshes. The garments category include full and half-sleeved Tshirts, Trousers, half-pants, kurta, dress, open shirt etc.

4.6.2 Analysis of Isomap Embeddings

We propose a novel strategy that allows establishing correspondences between different human scans , garments, or anything that resembles human body structure. SMPL being parametric human body model, acts as a reasonable medium to establish correspondences across different body shapes, poses and appearances. As explained in the main draft, once both the garment and the target body (parametric or non-parametric) are registered with SMPL, where the target body can be an SMPL mesh itself, we compute 128 dimensional isomap embeddings for each vertex of the garment and target body. Then, dense correspondences can be established between the two by matching similar 128-dimensional extrapolated features.

We arrive at this choice of feature modeling after carefully studying existing representations for dense correspondence matching for humans. This problem is specifically tough as humans are deformable objects and tend to undergo non-rigid motion. Continuous Surface Embeddings (CSE)[42] propose a learnable image-based representation of dense correspondences and a model which predicts, for each pixel in a 2D image, an embedding vector of the corresponding vertex in the object mesh, therefore establishing dense correspondences between image pixels and 3D object geometry. The authors show remarkable results in matching correspondences across RGB human images via 16-dimensional representation vectors. Recently, BodyMap[28] proposed to extend this approach by extrapolating the CSE embeddings of SMPLs registered with high-quality human scans in UV space. We started with BodyMap representation but later found it to produce a lot of false matching, and we decided to analyze the behavior quantitatively.

The representation for correspondence estimation should be rich and varied enough to avoid repetitions in the feature space when extrapolated, otherwise, different body parts would map nearby in the embedding space. More specifically, geodesically far-apart vertices should map far apart in the embedding space and vice-versa. Based on this ideation, we design an evaluation metric, **Richness Score**(\mathcal{R}_{score}) for each vertex v_i of SMPL mesh, which is calculated as follows:

$$\mathcal{R}_{score_i} = (\mathcal{R}_{near_i} + \mathcal{R}_{far_i})/2 \tag{4.15}$$

$$\mathcal{R}_{near_i} = \frac{1}{k^2} \sum_{i=1}^{k} min(|\mathcal{N}_{geo}^{rank} - \mathcal{N}_{emb}^{rank}|, k)$$
(4.16)



Figure 4.19: *Topwear:* The figure shows visualization of our collected dataset, first three rows depict the geometry of our collected garment in different poses, while last three shows the textured rendering of the respective geometries.



Figure 4.20: *BottomWear:* The figure shows visualization of our collected dataset, first three rows depict the geometry of our collected garment in different poses, while last three shows the textured rendering of the respective geometries.

$$\mathcal{R}_{far_i} = \frac{1}{k^2} \sum_{i=1}^{k} min(|\mathcal{F}_{geo}^{rank} - \mathcal{F}_{emb}^{rank}|, k)$$
(4.17)

where, \mathcal{N}_{geo}^{rank} & \mathcal{N}_{emb}^{rank} denotes the ranks of k-nearest neighbors of v_i in both geodesic and embedding space, and similarly, \mathcal{F}_{geo}^{rank} & \mathcal{F}_{emb}^{rank} denotes the ranks of k-farthest neighbors of v_i in both geodesic and embedding space. Thus, $\mathcal{R}_s core$ penalizes if the rank of neighbors (k-nearest and kfarthest) in geodesic and embedding space doesn't match. We report the values in Table 4.6, where it can be seen that extrapolating isoembedding values in Euclidean space has better effect than BodyMap[28]. The remaining values show that high dimensionality is preferred. However, empirically, values are saturated once a significant dimensionality is reached.

Table 4.5: Quantitative evaluation of WrinkleGeneration Network

3DHUMANS DATASET					
TYPE	P2S↓	ED↓	NC↑		
	x 10 ⁻³				
TW	1.156	2.362	0.8447		
BW	0.744	1.077	0.945		
THUMAN2.0					
TW	1.703	2.632	0.865		
BW	1.200	2.264	0.9159		

Table 4.6: Analysis of choice of representations for correspondence estimation. \mathcal{R}_{score} takes values between 0 & 1, where lower values are preferred.

Representation	$\mathcal{R}_{score}\downarrow$
BodyMap[28]	0.955
16-dim. Isomap Embeddings	0.491
32-dim. Isomap Embeddings	0.473
64-dim. Isomap Embeddings	0.437
128-dim. Isomap Embeddings	0.426
256-dim. Isomap Embeddings	0.424

4.6.3 Applications of the Proposed Framework

- **3D VTON for Arbitrary Garments** Our propose framework can be seen as a potential solution for 3D VTON problem. As evident from our qualitative results, the proposed framework can generalize well to unseen real and non-parametric garments, and retarget them to arbitrarily posed and shaped human scans.
- Size-fitting Solutions It is important to note that although we aim to preserve the overall structure of the garment to be retargeted, the final garment could scale accordingly to the target body. This is actually preferred as different people wear different sizes (M, L, XL, XXL) of the garments of the same style. Our framework can drape garments to arbitrary sizes (need not to be discreet) which is a unique contribution to the size-fitting solution.



Figure 4.21: The figure shows different real scanned garments of our *Dress Me Up* dataset draped onto real-scans of T-humans2.0 human body scans, (a) shows the *Dress Me Up*'s real-garments and columns (b) and (d) show scanned humans of Thumans2.0, we employ our proposed framework to drape these real garments to arbitrary real body scans of Thumans2.0 dataset as visualized in columns (c) and (e).

- Layered Clothing: As can be seen from our qualitative results on real scan, we can easily retarget garments on top of humans already wearing garments, thereby enabling layered clothing which is an extremely challenging task.
- Generating Ground Truth Data for 2D VTON Methods Since, we can retarget the 3D garment into different poses and even on different subjects, and eventually can render them consistently in form of 2D images, our framework can easily be used for generating photorealistic high-quality 2D VTON datasets from a limited number of 3D data samples. This is another highly useful application of our framework, and we intend to use it to develop and release such large-scale datasets in the public domain to accelerate the 2D VTON research as well.

4.6.4 Limitations & Future Work

We proposed a method for self-supervised 3D garment retargeting, and a first-of-its kind 3D VTON dataset for evaluating our framework. We showed that our novel framework leverages the isomap via SMPL to establish dense correspondences and initial coarse retargeting, which is then used as a prior

for training a self-supervised learning technique for refining the retargeting. Being the first method for retargeting (not just neural rendering) the 3D non-paramteric garment mesh from real-world distribution, we qualitatively show superior performance to similar State-of-the-Art methods.

Although we can retarget 3D garments on top of arbitrary human scans, currently there is no provision to remove the underlying garment the subject is already wearing. However, this is an extremely complex task as it might require reconstructing the underlying human body (for e.g. if a half t-shirt is to be draped over a subject wearing full t-shirt, removing full t-shirt requires reconstructing the arms of the subject). Though, we can easily handle noisy SMPL registration, small penetration noise can be noticed when the geometry of the input garment is bad, especially when the garment is reconstructed from RGB image using off-the-shelf networks (e.g. [73]). Finally, we aim to model extremely loose and free-flowing garments, such as long gowns, *sarees*, etc. We hope our method paves the way for handling the aforementioned problems we would like to tackle in future.

Chapter 5

Conclusion

In this chapter, we provide a brief conclusion on what we have achieved so far and the impact of the proposed methods. We also discuss the potential future direction that can be explored.

5.1 Discussion

In this thesis, we explored self-supervised methods for applications in 3D computer vision which overcome the major limitations of supervised methods. In particular, we find solutions to the problem of UV parameterization of general objects and garment retargeting in the 3D setting.

First, we designed a self-supervised framework for surface parameterization with both open and closed surfaces. We divide the closed surface into multiple patches leveraging a diffusion-enabled encoder that captures global-to-local context. The extracted patches, like open and homogenous to disk and are individually parameterized by learning forward and backward UV mapping for individual patches. Our framework is also discretization agnostic, enabling inference on meshes of arbitrary resolution. We show significant improvement in inference time and provide a comparison with existing parameterization on SHREC11 [35] dataset. Some of the limitations of the current method are that they fail to parameterise objects with very high extrinsic curvature, can sometimes have disconnected patches and overlaps. These can be overcome by designing better losses and constraints.

Secondly, we propose a self-supervised method for draping non-parameterized, 3D garment meshes over human body meshes of arbitrary shapes and poses. We first obtain initial alignment between the garment and the human body by establishing correspondences via Isomap Embeddings. We further refine this coarse retargeting by training an MLP that preserves the geometry of the garments guided by our novel losses. We propose a wrinkle generation module to obtain realistic details on the draped garments. We also contribute a new dataset of real-world reposed garments with realistic noise and topological deformations. However, the current method is highly dependent on correspondences, is not temporally consistent and does not generalise to loose clothing. These are some of the limitations to be targetted in future works.

5.2 Impact

This thesis advances the filed of virtual try-on systems significantly while providing a learning-based solution for parameterization. The proposed methods enable leveraging publicly available data via self-supervised methods. The proposed methods can be used in several applications in building a Metaverse. UV parameterization can be further extended to be used in creating, editing, and personalization of digital avatars. From the business point of view, Garment retargeting has a huge application in the eCommerce industry, enabling increased sales by reducing returns while saving cost and time. It can help enhance the online shopping experience to a great extent by facilitating the personalization and customization of garments. To our best knowledge, we are the first ones to provide a real-world dataset of the same garments in different poses, setting a benchmark for other future works.

5.3 Future Directions

This thesis paves the way for several explorations in the future. We elaborate on some of the potential research directions that can be pursued based on this thesis.

- **Differentiable Parameterisation:** Our current method performs seam estimation and parameterization separately, which is achieved by modules that are individually differentiable. However, combining both into an end-to-end differentiable pipeline allows it to be used as a plugin for learning both parameterization and texture map together in methods like [22], [23].
- Learning Texture from images: The UV parameterization method can be extended to make it controllable so that all the UVs of similar objects are mapped to the same UV space. This will further allow us to learn a common texture map for similar objects. This work can find applications that require editing large meshes number of meshes together; like in character modeling or editing.
- **Implicit Learning on garments:** Our current method of garment retargeting uses explicit representation of the 3D surface by mapping the input mesh to desired retargeted output. Instead, learning to map the mesh to an implicit representation instead of mesh will help leverage its advantages, such as flexibility in mesh resolution, memory efficiency, etc.
- Generalisation to loose clothing: Our current garment retargeting method only works for reasonably tight clothing. This method can be further extended for retargeting loose clothing by learning skinning weights of garments.

Publications

Thesis Publications

- Shanthika Naik, Chandradeep Pokhariya, Astitva Srivastava, Avinash Sharma; *Discretization-Agnostic Deep Self-Supervised 3D Surface Parameterization*; SIGGRAPH Asia - Technical Communications, 2022. (Patent application under submission.)
- Shanthika Naik, Kunwar Singh, Astitva Srivastava, Dhawal Sirikonda, Amit Raj, Varun Jampani and Avinash Sharma; *Dress Me Up: A Dataset & Method for Self-Supervised 3D Garment Retargeting*; Under review at International Conference on Computer Vision(ICCV) 2023.

Other Publications

 Shanthika Naik, Aryamaan Jain, Avinash Sharma, KS Rajan; Deep Generative Framework for Interactive 3D Terrain Authoring and Manipulation; In IGARSS 2022 - 2022
 IEEE International Geoscience and Remote Sensing Symposium 2022.
Bibliography

- N. Aigerman, K. Gupta, V. G. Kim, S. Chaudhuri, J. Saito, and T. Groueix. Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes, 2022.
- [2] B. AlBahar, J. Lu, J. Yang, Z. Shu, E. Shechtman, and J.-B. Huang. Pose with Style: Detail-preserving pose-guided image synthesis with conditional stylegan. *ACM Transactions on Graphics*, 2021.
- [3] T. Alldieck, H. Xu, and C. Sminchisescu. imghum: Implicit generative models of 3d human shape and articulated pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5461–5470, 2021.
- [4] H. Bertiche, M. Madadi, and S. Escalera. Cloth3d: Clothed 3d humans. In European Conference on Computer Vision, pages 344–359. Springer, 2020.
- [5] H. Bertiche, M. Madadi, and S. Escalera. Deep parametric surfaces for 3d outfit reconstruction from single view image. In 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), pages 1–8, 2021.
- [6] H. Bertiche, M. Madadi, and S. Escalera. Neural implicit surfaces for efficient and accurate collisions in physically based simulations. *CoRR*, abs/2110.01614, 2021.
- [7] H. Bertiche, M. Madadi, and S. Escalera. Pbns: Physically based neural simulation for unsupervised garment pose space deformation. *ACM Trans. Graph.*, 40(6), dec 2021.
- [8] H. Bertiche, M. Madadi, and S. Escalera. Pbns: Physically based neural simulation for unsupervised garment pose space deformation. ACM Trans. Graph., 40(6), dec 2021.
- [9] H. Bertiche, M. Madadi, and S. Escalera. Neural cloth simulation. ACM Trans. Graph., 41(6), nov 2022.
- [10] H. Bertiche, M. Madadi, E. Tylson, and S. Escalera. Deepsd: Automatic deep skinning and pose space deformation for 3d garment animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5471–5480, 2021.
- [11] B. L. Bhatnagar, G. Tiwari, C. Theobalt, and G. Pons-Moll. Multi-Garment Net: Learning to dress 3D people from images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [12] Z. Chen, K. Yin, and S. Fidler. Auv-net: Learning aligned uv maps for texture transfer and synthesis, 2022.
- [13] S. Choi, S. Park, M. Lee, and J. Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2021.

- [14] E. Corona, A. Pumarola, G. Alenyà, G. Pons-Moll, and F. Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In *CVPR*, 2021.
- [15] CVIT. 3dhumans: A rich 3d dataset of scanned humans, 2021.
- [16] L. De Luigi, R. Li, B. Guillard, M. Salzmann, and P. Fua. Drapenet: Generating garments and draping them with self-supervision, 2022.
- [17] D. A. Field. Laplacian smoothing and delaunay triangulations. *Communications in applied numerical methods*, 4(6):709–712, 1988.
- [18] V. Gabeur, J.-S. Franco, X. Martin, C. Schmid, and G. Rogez. Moulding humans: Non-parametric 3D human shape estimation from single images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [19] A. Grigorev, B. Thomaszewski, M. J. Black, and O. Hilliges. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2023.
- [20] T. Groueix, M. Fisher, V. G. Kim, B. Russell, and M. Aubry. AtlasNet: A papier-mâché approach to learning 3D surface generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [21] Y. Guo, X. Chen, B. Zhou, and Q. Zhao. Clothed and naked human shapes estimation from a single image. In *International Conference on Computational Visual Media*, pages 43–50. Springer, 2012.
- [22] M. Habermann, L. Liu, W. Xu, G. Pons-Moll, M. Zollhoefer, and C. Theobalt. Hdhumans: A hybrid approach for high-fidelity digital humans, 2022.
- [23] M. Habermann, L. Liu, W. Xu, M. Zollhoefer, G. Pons-Moll, and C. Theobalt. Real-time deep dynamic characters. ACM Trans. Graph., 40(4), jul 2021.
- [24] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis. Viton: An image-based virtual try-on network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7543–7552, 2018.
- [25] T. He, J. Collomosse, H. Jin, and S. Soatto. Geo-PIFu: Geometry and pixel aligned implicit functions for single-view human reconstruction. In Advances in Neural Information Processing Systems (NeurIPS), 2020.
- [26] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.
- [27] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung. ARCH: Animatable reconstruction of clothed humans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
- [28] A. Ianina, N. Sarafianos, Y. Xu, I. Rocco, and T. Tung. Bodymap: Learning full-body dense correspondence map. In CVPR, 2022.
- [29] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

- [30] V. Jampani, M. Kiefel, and P. V. Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [31] B. Jiang, J. Zhang, Y. Hong, J. Luo, L. Liu, and H. Bao. Benet: Learning body and cloth shape from a single image. ArXiv, abs/2004.00214, 2020.
- [32] S. S. Jinka, A. Srivastava, C. Pokhariya, A. Sharma, and P. J. Narayanan. Sharp: Shape-aware reconstruction of people in loose clothing. *International Journal of Computer Vision*, Dec. 2022.
- [33] S. Lee, G. Gu, S. Park, S. Choi, and J. Choo. High-resolution virtual try-on with misalignment and occlusion-handled conditions. arXiv preprint arXiv:2206.14180, 2022.
- [34] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. ACM Trans. Graph., 21(3):362–371, jul 2002.
- [35] B. Li, A. Godil, M. Aono, X. Bai, T. Furuya, L. Li, R. J. López-Sastre, H. Johan, R. Ohbuchi, C. Redondo-Cabrera, et al. Shrec'12 track: Generic 3d shape retrieval. In *Proceedings of Eurographics Workshop on* 3D Object Retrieval (3DOR), pages 119–126, 2012.
- [36] M. Li, D. M. Kaufman, V. G. Kim, J. Solomon, and A. Sheffer. Optcuts: Joint optimization of surface cuts and parameterization. ACM Transactions on Graphics, 37(6), 2018.
- [37] R. Li, B. Guillard, E. Remelli, and P. Fua. Dig: Draping implicit garment over the human body, 2022.
- [38] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. ACM Transactions on Graphics (ToG), 2015.
- [39] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019.
- [40] S. Majithia, S. N. Parameswaran, S. Babar, V. Garg, A. Srivastava, and A. Sharma. Robust 3d garment digitization from monocular 2d images for 3d virtual try-on systems. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3428–3438, 2022.
- [41] A. Mir, T. Alldieck, and G. Pons-Moll. Learning to transfer texture from clothing images to 3d humans. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, jun 2020.
- [42] N. Neverova, D. Novotný, V. Khalidov, M. Szafraniec, P. Labatut, and A. Vedaldi. Continuous surface embeddings. ArXiv, abs/2011.12438, 2020.
- [43] C. Patel, Z. Liao, and G. Pons-Moll. TailorNet: Predicting clothing in 3D as a function of human pose, shape and garment style. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2020.
- [44] A. Pumarola, J. Sanchez, G. Choi, A. Sanfeliu, and F. Moreno-Noguer. 3DPeople: Modeling the Geometry of Dressed Humans. In *International Conference in Computer Vision (ICCV)*, 2019.
- [45] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2016.

- [46] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference* on Computer Vision (ICCV), 2019.
- [47] S. Saito, T. Simon, J. Saragih, and H. Joo. PIFuHD: Multi-level pixel-aligned implicit function for highresolution 3D human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [48] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, page 409–416, New York, NY, USA, 2001. Association for Computing Machinery.
- [49] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, page 409–416, New York, NY, USA, 2001. Association for Computing Machinery.
- [50] I. Santesteban, M. A. Otaduy, and D. Casas. SNUG: Self-Supervised Neural Dynamic Garments. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [51] R. Sawhney and K. Crane. Boundary first flattening. ACM Trans. Graph., 37(1), dec 2017.
- [52] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov. Diffusion is all you need for learning on surfaces. *CoRR*, abs/2012.00888, 2020.
- [53] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces, 2020.
- [54] D. Song, T. Li, Z. Mao, and A.-A. Liu. Sp-viton: shape-preserving image-based virtual try-on network. *Multimedia Tools and Applications*, 79(45):33757–33769, 2020.
- [55] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pages 175–184, 2004.
- [56] G. Sperl, R. Narain, and C. Wojtan. Homogenized yarn-level cloth. ACM Transactions on Graphics (TOG), 39(4), 2020.
- [57] G. Sperl, R. Narain, and C. Wojtan. Mechanics-aware deformation of yarn pattern geometry. *ACM Transactions on Graphics (TOG)*, 40(4), 2021.
- [58] A. Srivastava, C. Pokhariya, S. S. Jinka, and A. Sharma. Xcloth: Extracting template-free textured 3d clothes from a monocular image. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 2504–2512, New York, NY, USA, 2022. Association for Computing Machinery.
- [59] vchoutas. https://github.com/vchoutas/smplify-x, 2019.
- [60] R. Vidaurre, I. Santesteban, E. Garces, and D. Casas. Fully Convolutional Graph Neural Networks for Parametric Virtual Try-On. *Computer Graphics Forum (Proc. SCA)*, 2020.

- [61] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang. Toward characteristic-preserving imagebased virtual try-on network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 589–604, 2018.
- [62] H. Wang, K. A. Sidorov, P. Sandilands, and T. Komura. Harmonic parameterization by electrostatics. ACM Trans. Graph., 32(5), oct 2013.
- [63] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo. Deep geometric prior for surface reconstruction, 2018.
- [64] Y. Xiu, J. Yang, X. Cao, D. Tzionas, and M. J. Black. ECON: Explicit Clothed humans Obtained from Normals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.
- [65] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black. ICON: Implicit Clothed humans Obtained from Normals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13296–13306, June 2022.
- [66] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6183–6192, 2020.
- [67] R. Yu, X. Wang, and X. Xie. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10511– 10520, 2019.
- [68] T. Yu, Z. Zheng, K. Guo, P. Liu, Q. Dai, and Y. Liu. Function4d: Real-time human volumetric capture from very sparse consumer rgbd sensors. In *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR2021), June 2021.
- [69] F. Zhao, Z. Xie, M. Kampffmeyer, H. Dong, S. Han, T. Zheng, T. Zhang, and X. Liang. M3d-vton: A monocular-to-3d virtual try-on network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13239–13249, October 2021.
- [70] F. Zhao, Z. Xie, M. Kampffmeyer, H. Dong, S. Han, T. Zheng, T. Zhang, and X. Liang. M3d-vton: A monocular-to-3d virtual try-on network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13239–13249, 2021.
- [71] Z. Zheng, T. Yu, Y. Liu, and Q. Dai. Pamir: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [72] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu. DeepHuman: 3D human reconstruction from a single image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [73] H. Zhu, L. Qiu, Y. Qiu, and X. Han. Registering explicit to implicit: Towards high-fidelity garment mesh reconstruction from single images, 2022.

- [74] H. Zhu, X. Zuo, S. Wang, X. Cao, and R. Yang. Detailed human shape estimation from a single image by hierarchical mesh deformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4491–4500, 2019.
- [75] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):198–207, 2002.