Metaverse in the Wild: Modeling, Adapting, and Rendering of 3D Human Avatars from a Single Camera

A DISSERTATION PROPOSAL
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Jae Shin Yoon

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Hyun Soo Park

June, 2022

Acknowledgements

I would like to thank my advisor Prof. Hyun Soo Park for his wholehearted support on my PhD journey. His insights lead me to the right and straight path; his ambition and passion wake the dream that was nestled in the corner of my heart up; his honesty and humbleness inspire me to learn how to build strong, meaningful, and healthy relationship with the ones who I care; his patience and generosity guide a noivce meditator to practice being consistent in research; his flexibility and tolerance cultivate a traveler who can collaborate with the people in the world. I also would like thank Prof. Volkan Iser, Prof. Catherine Zhao, and Prof. Christian Theobalt as the committee members for their valuable time and comments on my thesis.

I am fortunate to experience many collaborations with wise people. I thank Dr. Takaaki Shiratori and Dr. Shoou-I Yu for giving me boundless trust on a fresh and clumsy PhD student at the Facebook Reality Labs; thank Dr. Kihwan Kim for teaching me how to enjoy the essence of uncertainty in research at the NVIDIA; thank Prof. Christian Theobalt for making me realize the power of creative culture at the MPII; and thank Dr. Duygu Ceylan for introducing me the power of soft leadership at the Adobe. I also thank all our group members for their help and valuable comments, including Zhixuan Yu, Yasamin Jafarian, Jingfan Guo, Jayant Sharma, Praneet Bala, Tien Do, Zachary Chavis, and Meng-Yu Jennifer Kuo.

Finally, I thank all my family and friends who always emotionally, physically, and financially support my PhD journey. In particular, I want to thank my friends, Jayant Gupta, Ryan Chan, and Feng Tian, who help with surviving my US life, and I strongly believe whatever achievement I made was not possible without the dedication of my wife, Jaekyung Yang.

Abstract

Metaverse is poised to enter our daily lives as new social media. One positive application would be tele-presence that allows users to interact with others through the photorealistic 3D avatars using AR/VR headsets. Such tele-presence requires high fidelity 3D avatars, depicting fine-grained appearance, e.g., pore, hair, wrinkle on face, from any viewpoint. Previous works have utilized a system of multiview cameras to generate the 3D avatars, which enables measuring appearance and 3D geometry of a subject. Deploying such large camera systems in our daily environment, however, is often difficult in practice due to the requirement of camera infrastructure with precisely controlled lighting. In this dissertation, I will **develop a computational model that can reconstruct a 3D human avatar from a single camera whose quality is equivalent to that from multi-camera system by learning from data.**

The main challenge for learning to reconstruct a 3D avatar from a single camera comes from the lack of 3D ground truth data. A distribution of human geometry and appearance is extremely diverse, depending on a number of parameters such as identity, shape (slim vs. fat), pose, apparel style, viewpoint, and illumination. While a data-driven model requires to learn from the data that can span such diversity, no such data exists to date. I address this challenge by developing a set of self-supervised algorithms that allow learning a generalizable visual representation of dynamic humans to reconstruct a 3D avatar from a single camera; to adapt the 3D avatar to unconstrained environment; and to render fine-grained appearance of the 3D avatar.

Learning to reconstruct a 3D avatar from a single view image. Large 3D ground truth data are required to learn a visual representation which describes the geometry and appearance of dynamic humans. I collect a large corpus of training data from a number of people using a multi-camera system which allows measuring a human with minimum occlusion. 107 synchronized HD cameras capture 772 subjects across gender, ethnicity, age, and garment style with assorted body poses. From the multiview image streams, I reconstruct 3D mesh models to represent human geometry and appearance without missing parts. By learning the images and reconstruction results, the AI model can generate a complete 3D avatar from a single view image.

Learning to adapt the learned 3D avatar to general unconstrained scenes. The quality of the learned 3D avatar is often degraded when the visual statistics of the testing data largely deviates from that of the training data, e.g., the lighting in the controlled lab environment (training) is very different from the unconstrained outside environment (testing). To mitigate such domain mismatch, I introduce a new learning algorithm that can adapt the learned 3D avatars to unconstrained scenes by enforcing the spatial and temporal appearance consistency, i.e., the appearance of the generated 3D avatar should be consistent with the one observed from the image of unconstrained scenes and the one generated from the previous time. Applying these consistency to a short sequence of testing images makes it possible to refine the visual representation without any 3D ground truth data, allowing to generate high-fidelity 3D avatars from everywhere.

Learning to render fine-grained appearance of the 3D avatars from diverse people. High quality geometry is the main requirement for fine-grained appearance rendering of a 3D avatar. However, the learned visual representation is designed to reconstruct such geometry only for the limited number of people (e.g., a single subject) due to the lack of 3D ground truth data, which no longer exists for other subjects out of training data. I bypass this problem by introducing a pose transfer network that learns to render fine-grained appearance without high quality geometry. Specifically, a pose encoder encodes the pose information from a 3D body model that represents the coarse surface geometry of general undressed humans, and an appearance decoder generates the fine-grained appearance (sharp 2D silhouette and detailed local texture) which is reflective of the encoded body pose for a specific subject seen from a single image. We further embed the 3D motion representation to the encoder in a form of temporal derivatives of 3D body models observed from a video, which allows the decoder to augment the physical plausibility by rendering the motion-dependent texture, i.e., wrinkle and shade on the clothing that are motivated by human movements. Eliminating the requirement of the high quality geometry brings out strong generalization of the rendering model to anybody from a single image or video.

In the experiment, I demonstrate that the reconstructed 3D avatar is accurate and temporally smooth; the learned visual representation is highly generalizable to diverse scenes and people; and the rendering results of the 3D avatars is photorealistic compared

to previous 3D human modeling and rendering methods. Beyond social tele-presence, enabling various applications is also possible: I apply the learned human visual representation to creating bullet time effect, image relighting, virtual navigation of a 3D scene with people, motion transfer and video generation from a still image.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Figure 1.1. (Left): For virtual interactions with other people, we use passive 2D video of a single camera which does not provide an active control of viewpoint of others [5]. (Right): A specialized capturing equipment such as a number of cameras with precisely controlled lighting can produce production-level quality of 3D avatars whose geometry and appearance are perceptually indistinguishable from the real person, enabling the active viewpoint control of others [6]. This dissertation will bridge the gap between them by introducing a new AI model that can reconstruct production-level quality of 3D human avatars from a single camera.

Virtual social interactions are becoming deeply integrated into our daily lives, in particular, under the impact of COVID-19, allowing social distancing yet staying connected. So far, we still rely predominantly on passive 2D videos as a communication tool as shown in Figure 1.1. This makes a sharp contrast with our physical social interactions occurred in 3D where we actively move our body to see others from different views.

(a) Illumination & scene (b) Pose & view (c) Identity

Figure 1.2. Diverse human and scene nature. A distribution of human appearance is extremely diverse, dependent on a number of parameters such as (a) illumination, scene, (b) pose, viewpoint, and (c) identity including fashion and hair style, skin color, facial structure, and body shape.

To enable such real interactions in virtual space, we need production-level quality of 3D avatars whose appearance is perceptually and geometrically indistinguishable from the real person, which makes it possible to observe others from any viewpoints as we move. Early works have attempted to reconstruct such 3D avatars by leveraging massive hardware such as a hundred of multiple cameras with precisely controlled synchronization and lighting systems as shown in Figure 1.1. However, deploying such large camera rigs and expensive control systems into our daily lives is not practical. The goal of this dissertation is **to develop a computational model that can reconstruct a 3D avatar from an image or video of a monocular camera whose quality is equivalent to that made from a multi-camera system by learning from data.** This single view based 3D avatar reconstruction will be the foundation for numerous real-world applications such as 3D virtual clothing try-on for online shopping, self-avatar reconstruction for gaming, behavioral monitoring for children and elderly care, and human-robot interaction for social service system.

Dataset	# of subjects	Measurement method	Face	Hand	Body	Cloth
FACS [29]	41	2 cameras (passive stereo)	✓			
FaceWarehouse [30]	150	RGBD Microsoft Kinect	✓			
CMU Multi-PIE [31]	337	15 cameras	✓			
3DMM [32]	200	3D scanner	✓			
4DFAB [33]	180	1 RGBD Kinect camera, 1 stereo (2 cameras), and 1 frontal camera	✓			
BFM [34]	200	3D scanner	✓			
ICL [35]	10,000	3D scanner	✓			
FaceScape [36]	938	68 DSLR cameras	✓			
NYU Hand [37]	2 (81K samples)	Depth camera		✓		
HandNet [38]	10 (213K samples)	Depth camera and magnetic sensor		✓		
BigHand 2.2M [39]	10 (2.2M samples)	Depth camera and magnetic sensor		✓		
RHD [40]	20 (44K samples)	N/A (synthesized)		✓		
STB [41]	1 (18K samples)	1 pair of stereo cameras		✓		
FreiHand [42]	N/A (33K samples)	8 cameras		✓		
CMU Mocap	∼100	Marker-based			✓	
CMU Skin Mocap [43]	<10	Marker-based	✓		✓	
INRIA [11]	N/A	Markerless (34 cameras)			✓	✓(natural)
Human EVA [44]	4	Marker-based and Markerless (4-7 cameras)			✓	
Human 3.6M [14]	11	Markerless (depth camera and 4 HD cameras)			✓	
Panoptic Studio [7, 45]	∼100	Markerless (31 HD and 480 VGA cameras)		✓	✓	
Dyna [9]	10	Markerless (22 pairs of stereo cameras)			✓	
ClothCap [10]	10	Markerless (22 pairs of stereo cameras)				✓(synthesized)
BUFF [46]	5	Markerless (22 pairs of stereo cameras)			✓	✓(natural)
3DPW [47]	7	Marker-based (17 IMUs) and Markerless (1 camera + 3D scanner)			✓	✓(natural)
TNT15 [48]	4	Marker-based (10 IMUs) and Markerless (8 cameras + 3D scanner)			✓	
D-FAUST[49]	10	Markerless (22 pairs of stereo cameras)			✓	
HUMBI (ours)	772	Markerless (107 HD cameras)	✓	✓	✓	✓(natural)

Table 1.1. The summary of existing 3D ground truth datasets for human body expressions including gaze, face, hand, body, and clothing.

1.1 Challenge: Lack of 3D Ground Truth Data

We have witnessed a remarkable progress on many AI models, e.g., convolutional neural networks, for various computer vision tasks such as detection [50, 51], recognition [52, 53, 54], and segmentation [55, 56] of which the performance is often driven by the quality and quantity of the training dataset. The AIs of 3D avatars are not exceptional: they require numerous human visual data to learn. However, collecting the 3D ground truth data that describe the geometry and appearance of every possible human is very difficult because its distribution is extremely diverse, depending on a number of parameters such as identity, shape (slim vs. fat), apparel style, viewpoint, illumination, and pose as shown in Figure 1.2 where the existing 3D ground truth data sets, e.g., a hundred of subjects for body and clothing as summarized in Table 1.1, lack such diversity. The AI models learned from such limited datasets have shown very weak generalizability, e.g., 3D avatar reconstruction is possible only for a single subject from a specific scene [57, 58], whose application to unseen people and scenes produces significant visual artifacts such

as unrealistic shape and appearance.

1.2 Our Approach

The lack of 3D ground truth data is the core challenge for data-driven 3D avatar reconstruction from a single camera as described in Section 1.1. We address this challenge by introducing a set of self-supervised learning algorithms which enable us to develop a generalizable visual representation of geometry and appearance of dynamic humans to reconstruct a 3D avatar from a single camera (Part I); to adapt the learned 3D avatar to unconstrained scenes (Part II); and to render fine-grained appearance of 3D avatars of diverse people (Part III). The overview of our representation learning pipeline is visualized in Figure 1.3.

In Part I, we formulate a scene- and person-specific visual representation to reconstruct a 3D avatar from a single camera by learning large amount of 3D ground truth data. This representation is modeled by a three-step pipeline of capturing, measuring, and learning of dynamic humans using the system of synchronized multiple cameras.

In Part II, we generalize the learned visual representation to unconstrained scenes with our scene-agnostic domain adaptation framework. In particular, we use the temporal structure in data from a video of unconstrained scenes as a source of self-supervisory signal for the adaptation of the learned 3D avatar without any 3D ground truth data.

In Part III, we generalize the visual representation to diverse people by developing a person-agnostic generative neural network that learns to render fine-grained appearance even from imperfect geometry reconstruction of a person from a single image or video.

1.2.1 Part I: Learning to Reconstruct 3D Avatars from a Single Camera

In this part, our goal is to develop an AI model that can learn a visual representation of dynamic humans to reconstruct a high fidelity 3D avatar from a single camera by learning large amount of 3D ground truth data. To enable such AI model, we utilize a multi-camera system to capture large amount of images of assorted people from dense camera viewpoints (Chapter 3); to reconstruct the 3D ground truth data from the captured images (Chapter 3 and 4); and to train a convolutional neural network that

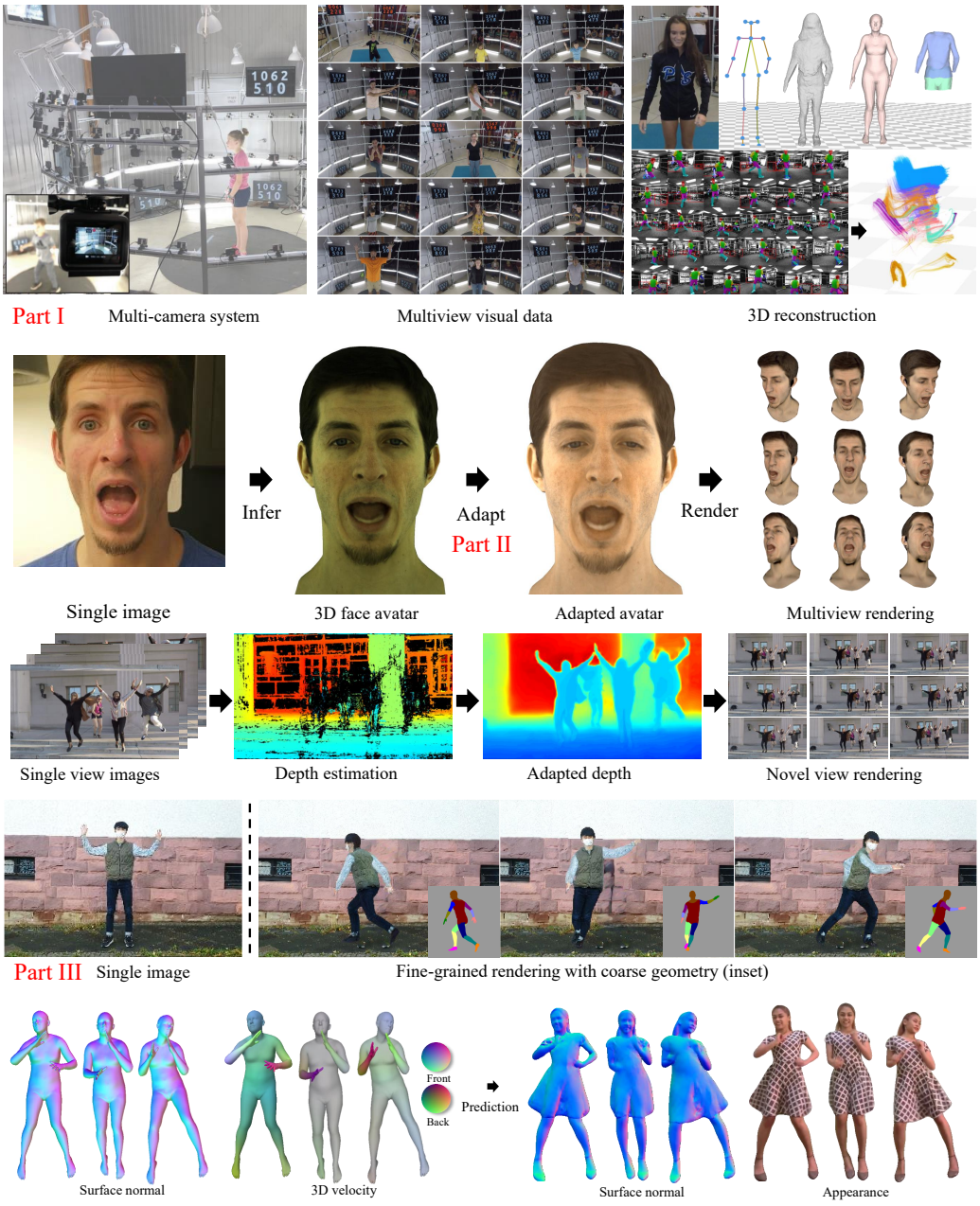


Figure 1.3. The overview of our representation learning pipeline. The place where the new part starts is marked in red.

can infer a 3D avatar from a single image by learning the reconstructed 3D ground truth data (Chapter 5).

In Chapter 3, we presents a large multiview dataset for human body expressions under natural clothing. 107 synchronized HD cameras are used to capture 772 distinctive subjects across gender, ethnicity, age, and physical condition, resulting in collecting $\sim 100 \times 10^6$ multiview images. From the multiview image streams, we reconstruct 3D mesh models to represent human geometry and appearance without missing parts. Using these dataset, we formulate a new benchmark challenge of a pose-guided appearance rendering task that aims to substantially extend photorealism in modeling diverse human expressions in 3D.

In Chapter 4, we introduce a 3D clustering algorithm for semantic segmentation of 3D point clouds. Given a 3D point underlying human body surface, we build a 3D semantic map which represents the probability distribution over body parts constructed by a set of 2D semantic recognition across multiple views. This semantic map is further augmented in the time domain with local motion priors where the 3D points that belong to the same body part will undergo similar local rigid transformation. The augmented semantic map allows the clustering of the 3D point clouds into six body parts (i.e., head, torso, and upper-/lower-arms, upper-/lower-legs) in a temporally coherent way, providing the 3D ground truth data for part-specific geometry.

In Chapter 5, we develop a novel convolutional neural network that can learn a visual representation from large 3D ground truth data to infer a 3D avatar from a single image. To enable this, we newly design a multimodal autoencoder which outputs a complete 3D surface geometry and appearance from an input 2D image. This single view reconstructed 3D avatar enables multiview rendering of photorealistic humans.

1.2.2 Part II: Learning to Adapt the Learned 3D Avatars to General Unconstrained Scenes

Our visual representation is designed to learn large 3D ground truth data (i.e., images and 3D reconstruction results) to generate a high fidelity 3D avatar from a single image. However, the learned representation often does not perform well when the visual statistics of the input images of testing data highly deviates from that of training data, e.g., the lighting of the controlled lab environment (training) is very different from that of

the unconstrained outside environment (testing), leading to generating low quality 3D avatar with distorted appearance and geometry. To address this domain mismatch, we introduce a set of self-supervised learning algorithms to adapt the visual representation to unconstrained general scenes by utilizing the temporal observation from a video, i.e., a sequential set of images, where no 3D ground truth data is required.

In Chapter 6, we adapt the visual representation to the scenes under the illumination changes by enforcing the spatial and temporal appearance consistency. Spatial consistency is used to refine the visual representation in a way that minimizes the color differences between the appearance of the generated 3D avatars and the one observed from the images of unconstrained scenes. Temporal consistency is designed to improve the temporal coherence of the visual representation by minimizing the color differences of the 3D avatars generated from the images of consecutive times. Applying these consistency to a short sequence of testing images brings out high fidelity 3D avatars everywhere.

In Chapter 7, we adapt the visual representation to the scenes under the camera viewpoint changes by enforcing view-invariant 3D motion consistency. Inspired by the human motion nature, i.e., smooth and slow [59], we formulate the linear relationship of the 3D body motion seen from different camera viewpoints. Under this formulation, the learned visual representation is constrained to predict the coherent geometry of the 3D avatars from the images of a single moving camera, enabling geometrically plausible novel view synthesis.

1.2.3 Part III: Learning to Render Fine-Grained Appearance of 3D Avatars of Diverse People

In Part III, we study the visual representation for appearance rendering of diverse people. In particular, we focus on developing a generative rendering model that can synthesize fine-grained appearance without high quality geometry which is the main requirement for human rendering in many previous works [60, 61, 62]. Eliminating such requirement brings out strong generalization of the rendering model to anybody from a single image.

In Chapter 8, we present a new design of compositional generative networks that predict the silhouette, garment labels, and texture as a function of a 3D body model

that describes the coarse surface geometry of general undressed humans. Each modular network is explicitly dedicated to a subtask that can be learned without ground truth geometry data. At the inference time, we utilize the trained network to produce a unified map of appearance and its labels in UV coordinates, which remains constant across body poses. The unified map provides an incomplete yet strong guidance to generating the appearance in response to the pose changes. Finally, we use the trained network to complete the fine-grained appearance.

In Chapter 9, we present a compact 3D motion representation by enforcing equivariance—a representation is expected to be transformed in the way that the pose is transformed. We model an equivariant encoder that can generate the generalizable representation from the spatial and temporal derivatives of the 3D body surface. This learned representation is decoded by a compositional multi-task decoder that renders high fidelity time-varying appearance. Our experiments show that our method can generate a temporally coherent video of dynamic humans for unseen body poses and novel views given a single view video.

1.3 Validation

In this dissertation, we introduce a set of representation learning algorithms to reconstruct high fidelity 3D avatars from a single camera. To validate our pipeline, we perform extensive evaluation of our method in terms of accuracy, visual quality, generalizability, temporal stability, and applicability. For accuracy, we perform manual annotation of 2D keypoints and shape mask on the testing images and compare with the ones from the 2D reprojection of the reconstructed 3D avatars. For visual quality, we synthesize the images of the 3D avatars from a novel viewpoint and compute the perceptual similarity [63] with a real ground truth image. For generalizability, we demonstrate a consistent performance of our visual representation across scene, viewpoint, and identity changes. For temporal stability, we measure the temporal smoothness of the reconstructed 3D avatars over times and its standard deviation with respect to the entire video frames. For applicability, we show various applications such as cinemagraphs, bullet time effect, motion transfer, and relighting.

1.4 Contributions

This thesis introduces fundamentals of an AI based human modeling, adapting, and rendering technologies to formulate a generalizable visual representation for single view 3D avatar reconstruction. In summary, the contributions of this thesis as follows:

- A large multiview dataset and scalable 3D reconstruction algorithms to collect large 3D ground truth training data of geometry and view-specific appearance for diverse human body expressions.
- A novel multimodal autoencoder that can learn large 3D ground truth data to output a 3D avatar from an input 2D image.
- A set of self-supervised algorithms for domain adaptation of the learned 3D avatars to unconstrained scenes using temporal structure in data from the video of a monocular camera.
- A new generative approach to render fine-grained appearance from coarse surface geometry of general undressed humans.
- A new motion representation that allows to render high fidelity time-varying appearance by encoding spatial and temporal derivatives of 3D body models.

1.4.1 Publications

The relevant publication list for this thesis is as follows:

- (Chapter 3) HUMBI: A Large Multiview Dataset of Human Body Expressions [64], IEEE Computer Vision and Pattern Recognition (**CVPR**) 2020.
- (Chapter 3) HUMBI: A Large Multiview Dataset of Human Body Expressions and Benchmark Challenge [65], IEEE Transactions on Pattern Analysis and Machine Intelligence (**TPAMI**) 2022.
- (Chapter 4) 3D Semantic Trajectory Reconstruction from 3D Pixel Continuum [66], IEEE Computer Vision and Pattern Recognition (**CVPR**) 2018.

- (Chapter 5 and 6) Self-Supervised Adaptation of High-Fidelity Face Models for Monocular Performance Tracking [15], IEEE Computer Vision and Pattern Recognition (**CVPR**) 2019.
- (Chapter 7) Novel View Synthesis of Dynamic Scenes with Globally Coherent Depths from a Monocular Camera [67], IEEE Computer Vision and Pattern Recognition (**CVPR**) 2020.
- (Chapter 8) Pose-Guided Human Animation from a Single Image in the Wild [68], IEEE Computer Vision and Pattern Recognition (**CVPR**) 2021.
- (Chapter 9) Learning Motion-Dependent Appearance for High-Fidelity Rendering of Dynamic Humans from a Single Camera [69], IEEE Computer Vision and Pattern Recognition (**CVPR**) 2022.

Chapter 2

Related Work

Early research [6, 17, 9, 60] in graphics and computer vision for 3D human modeling have mainly focused on *how to reconstruct* high fidelity 3D avatars without restriction of the sensor measurement, whose geometry and appearance have been becoming indistinguishable from a real person. With the advent of deep neural networks [70, 70, 71, 72], this focus has been progressively shifted into *how to predict* such 3D avatars with the restriction of the sensor measurement, e.g., a single camera, which can be easily deployed in our everyday environment.

In this chapter, we review the literature which have introduced the innovations in 3D human modeling and rendering and relate them to our methods. In particular, we will go through the previous works that attempted various formulations of an AI-based visual representation for 3D humans and its generalization to diverse scenes and people.

2.1 Learning to Reconstruct 3D Avatar

An AI model requires large amount of 3D ground truth data to learn. In this section, we review existing works for capturing, measuring, and learning of dynamic humans to develop a data-driven visual representation which allows to predict 3D avatars from a single image.

2.1.1 Capturing Human Visual Dataset

Multiple camera infrastructure have been employed to capture large amount of human visual dataset. Some works utilized RGBD cameras, e.g., Microsoft Kinect, to produce high fidelity geometry with pore-level details [30, 33, 33, 73, 74, 30]. An infrared projector from the depth sensor physically measures the distance between the camera and human body surface, while RGB cameras capture the associated appearance (color and texture). However, the constructed geometry is largely incomplete due to the significant self-occlusion seen from a single sensor where the use of multiple depth sensors is highly limited by the interference between them.

A multi-camera system, i.e., a number of RGB cameras with synchronization systems, is a viable solution to overcome the challenges from a single camera that includes self-occlusion. Many existing works [29, 75, 36, 7, 14, 44, 49, 9, 42, 7, 29] have leveraged this system to capture human body expressions at high resolution with minimum self-occlusion. Given the multiview image streams, a complete 3D ground truth data can be obtained using a 3D reconstruction algorithm. For example, multiview stereo [76, 21] reconstructs the detailed surface geometry of human by performing per-pixel depth estimation from each view and unifying entire views in the coherent 3D space. Applying such reconstruction algorithms over time enables capturing the natural 3D clothing deformation in response to human body movements [77, 78, 10]. Notably, a multi-camera system has been used for 3D bootstrapping [45, 79] to annotate the 2D hand keypoints that are coherent to multiview images. Such large system is normally deployed in the fixed laboratory environment, limiting the diversity of the human visual dataset.

To capture diverse dataset from unconstrained environment, some existing works [80, 81, 82] have made an attempt to fit a computational 3D body model to a single image. For example, SMPL body model [17] is fitted to a single image based on the 2D annotation of body silhouette, keypoints, and parts segmentation [82]. Zhu et al. [81] fitted 3DMM [32] face model to 60K samples of human face images from several face alignment datasets [75, 83, 84, 85, 86]. Hampali et al. [80] introduced a large 3D hand dataset by fitting MANO [87] hand model to video sequences of double-handed object interaction. Notably, manually annotated image-to-surface correspondences on 50K COCO images [88] enabled a data-driven model that can detect human body surface from a single view image [82] without 3D body model fitting.

The diversity of the human visual dataset can be further augmented by synthetic data. In particular, graphics simulation produces numerous synthetic data by controlling various physical properties such as body shape, pose, lighting, background, appearance, garment style, and viewpoints. For instance, recent works [89, 2] simulate textured 3D body model [17] using pre-recorded motion archive [90, 91] and synthesize the 2D images by projecting the 3D model to virtual camera viewpoints under novel lighting condition and background scenes. Similar to this, generating synthetic hand dataset also follows the two-step pipeline of simulating the textured 3D hand model [87] and synthesizing the 2D images of the 3D hand model [92, 93, 94, 95].

Unlike existing datasets focusing on each body expression, our human visual dataset is designed to span appearance of total body expressions of face, body, hand, and clothing from a number of distinctive subjects using a dense camera array. Our mega-scale multiview visual data provide a new opportunity to develop a generalizable visual representation for pose- and view-specific appearance.

2.1.2 Measuring 3D Human Behavior

The pixels in a video can be tracked to form long term trajectories which are encoding the consistent semantic meaning across the time. We leverage such trajectory basis to measure human behavior in 3D, producing 3D ground truth data for part-specific geometry. We review existing works for 2D trajectory reconstruction and its extension to 3D.

2D Trajectory Reconstruction. As many objects are roughly rigid and move independently, motion provides a strong discriminative cue to group pixels and recognize occluding boundary, precisely. A core challenge of motion segmentation lies in fragmented nature of trajectories caused by tracking failure (occlusion, drifting, and motion blur). Embedding trajectories into low dimensional space has been used to robustly measure trajectory distance in the presence of missing data without pre-trained models [96, 97, 98, 99], and 2D trajectories can be decomposed into 3D camera motion and deformable object models [100, 101, 102]. Visual semantics learned by object recognition frameworks provides stronger cues to cluster trajectories [103, 104, 105].

3D Trajectory Reconstruction. Due to dimensional loss in the process of 2D projection, reconstructing 3D motion from a monocular camera is an ill-posed problem in general, i.e., the number of variables (3D motion parameters) is greater than the number equations (projections). However, when an object undergoes constrained deformation such as face, its 3D shape can be recovered by enforcing spatial regularity, e.g., shape basis [106, 107, 108, 109], template [110], and mesh [111]. A key challenge of this approach is to learn a shape prior that can express general deformation, often requiring an instance specific pre-trained model, or inherent rank minimization where the global solution is difficult to be achieved [112, 113]. A trajectory based representation directly addresses this challenge. Motion is described by a set of trajectory stream where generic temporal regularity is applied through DCT trajectory basis [114, 115], polynomial basis [116, 117], and linear dynamical model [118]. A spatiotemporal constraint can further reduce dimensionality, resulting in robust 3D reconstruction [100, 119, 120]. When multiple view images are used, it is possible to represent general motion with topological change without any spatial and temporal prior [121, 122].

Unlike 2D trajectories, semantic labeling of 3D trajectories is largely under-studied research area. Notably, Yan and Pollefeys [109] presented a trajectory clustering algorithm based on articulated body structure, i.e., an object is composed of a kinematic chain of rigid bodies where the articulated joint and its rotational axis lie in the intersection of two shape subspaces. Later, image segmentation cues have been incorporated to recognize a scene topology, i.e., pre-clustering object instances, to reconstruct dynamics scenes from videos in the wild [123, 124, 125]. Note that none of these work has addressed semantics. The work by Joo et al. [122] is closest to our approach where the trajectory clustering is based on 3D rigid transformation of human anatomical keypoints. Our method is not limited to human bodies, which enables modeling general human interactions with scenes, objects, and other people.

2.1.3 Representation Learning for Single View 3D Human Prediction

Our representation learning lies in the intersection between high fidelity appearance modeling and 3D model reconstruction from a monocular camera, which will be briefly reviewed in this section.

3D Face Modeling and Single View Reconstruction. Faces have underlying spatial structural patterns where low dimensional embedding can efficiently and compactly represent diverse facial configurations, shapes, and textures. Active Shape Models (ASMs) [126] have shown strong expressibility and flexibility to describe a variety of facial configurations by leveraging a set of facial landmarks. However, the nature of the sparse landmark dependency limits the reconstruction accuracy that is fundamentally bounded by the landmark localization. AAMs [127] address the limitation by exploiting the photometric measure using both shape and texture, resulting in compelling face tracking. Individual faces are combined into a single 3DMM [128] by computing dense correspondences based on optical flow in conjunction with the shape and texture priors in a linear subspace. Large-scale face scans (more than 10,000 people) from diverse population enables modeling of accurate distributions of faces [35, 129]. With the aid of multi-camera systems and deep neural networks, the limitation of the linear models can be overcome using Deep Appearance Models (DAMs) [60] that predicts high quality geometry and texture. Its latent representation is learned by a conditional variational autoencoder [130] that encodes view-dependent appearance from different viewpoints. Our approach eliminates the multi-camera requirement of the DAMs by adapting the networks to a video from a monocular camera.

The main benefit of the compact representation of 3D face modeling is that it allows estimating the face shape, appearance, and illumination parameters from a single view image. For instance, the latent representation of the 3DMMs can be recovered by jointly optimizing pixel intensity, edges and illumination (approximated by spherical harmonics) [131]. The recovered 3DMMs can be further refined to fit to a target face using a collection of photos [132] or depth based camera [133]. [134] leveraged expert designed rendering layers which model face shape, expression, and illumination and utilized inverse rendering to estimate a set of compact parameters which renders a face that best fits the input. This is often an simplification and cannot model all situations. In contrast, our method does not make any explicit assumptions on the lighting of the scene, and thus achieves more flexibility to different environments.

Other methods include [135, 136], which used cascaded CNNs which densely align the 3DMM with a 2D face in an iterative way based on facial landmarks. The geometry of a 3D face is regressed in a coarse-to-fine manner [137], and asymmetric loss enforces

the network to regress the identity consistent 3D face [138]. [139] utilizes jointly learned geometry and reflectance correctives to fit in-the-wild faces. [140] trained UV regression maps which jointly align with the 3DMM to directly reconstruct a 3D face.

Single View 3D Body and Clothing Reconstruction. Inspired by Johansson's point light display experiment [141], the spatial relationship of the human body has been actively studied to recover 3D humans [106, 142, 143]. For instance, diverse body poses and shapes can be modeled by a linear combination of blend shape basis [17]. Such models have been combined with body landmark recognition approaches using deep learning [144, 145, 146, 147], allowing an end-to-end 3D body model reconstruction from an image. To express diverse body poses originated from nonlinear articulated motion, various spatial biological constraints have been explored such as limb length constraint [119, 148], activity-specific pose [149], and a physical constraint (e.g., joint force and torque) [150, 151]. Recently, a large volume of literature has shown that the nonlinear manifold of human pose space can be effectively approximated by training data [152, 153, 154, 82, 89], which enables reasoning about a complete human body pose and shape from a single image.

Unlike the face and body, clothes reconstruction from a single image is particularly difficult because there is no off-the-shelf model that can represent their deformation. Further, the geometry is commonly very complex, e.g., having wrinkles, and recovering such details requires high resolution 3D measurements such as a multi-camera system [10, 155] or depth camera [156, 157]. RGBD imaging provides an opportunity to measure high resolution cloth at the wrinkle level [158, 156, 157, 159, 160]. Clothes reconstruction from a single RGB image has been studied in the context of nonrigid structure-from-motion [100, 161, 162, 156] with an assumption that the cloth deformation lies in a latent subspace where shape basis models can be learned online. However, such approaches model the cloth deformation in isolation where the physical interactions with the body surfaces are not taken into account. Clothes shape under such physical interactions can be reconstructed across different body shapes and poses [10, 163, 164] based on cloth-to-body 3D correspondences or manual deformation of a 2D image [165].

2.2 Learning to Adapt Visual Representation to General Unconstrained Scenes

For the widespread use of AI models, their learned visual representation should be generalizable and adaptive to diverse scenes. In this section, we study existing methods for the domain adaptation of human visual representation to unconstrained scenes and scene-agnostic representation learning.

2.2.1 Domain Adaptation of High Fidelity 3D Face Models

Human face is highly structured, e.g., symmetry, which provides various supervisory cues to generalize the learned priors. For example, the facial appearance should be consistent across the images from multiple viewpoints (multiview consistency), and to satisfy this, a neural network is required to learns to predict accurate 3D facial geometry of an unseen person by leveraging videos that captured facial performance [166]. Such multiview consistency can adapt a regression network that predicts low dimensional parameters (basis) of the 3D face model, reconstructing photo-realistic 3D facial geometry that is highly reflective of the real person from a single image [167]. Even a single face image can be a training source to refine the face priors: geometry, skin reflectance, and illumination can be jointly optimized by projecting the 3D model to a single image to match the visual statistics of the inferred 3D model with the real person [134, 139]. Chen et al. [168] adapts the face priors to a new environment in a self-supervised manner by enforcing appearance consistency between the 3D model and real person from the coherent UV coordinates that are invariant to the facial deformation. Our method extends this UV map based adaptation to the time domain (consecutive frames) for temporally stable face rendering of diverse people.

2.2.2 Scene-Agnostic Single View Depth Prediction

Single view depth estimation is highly ill-posed problem due to significant ambiguity, i.e., any 3D points on the camera ray can be a solution of the depth. Such ambiguity have been relaxed by formulating data-driven priors using massive amount of real-world data (the ground-truth pair of an image and associated depth map) [22, 169, 170] where a CNN model learns to predict the depth map by referring to the monocular

cues such as shading, vanishing points and occlusion. However, such cues are not consistent across the camera views and scenes. To adapt the depth prediction model to a novel scene and viewpoint without any labeled data, some works [171, 172, 173] utilized stereo images to enforce the left-right consistency with known camera motion, which are not viable solution in the single camera application. Humans are a special case of spatial constraints, which allow markerless motion capture from a monocular camera [174, 175, 176]. Tan et al. [177] leveraged such spatial constraints in a self-supervised learning framework to refine the human depth estimation across the video frames. While this method generalizes the depth estimation model using a single camera, it only consider the foreground parts such that the rendered human images cannot be naturally blended with background scenes. Unlike the explicit spatial priors, our work makes use of general geometric priors and motion constraint to reconstruct a complete and view-invariant geometry of human dynamic scenes, which allows us to generate geometrically plausible view synthesis results.

2.3 Learning to Render Fine-Grained Appearance with Generative Models

While high quality geometry is the main requirement for fine-grained appearance rendering of a 3D avatar, it is often not available due to the lack of 3D ground truth data. Many previous works bypass this problem by combining existing rendering frameworks with a generative model such as generative adversarial networks (GAN) [178] which allows synthesizing photorealistic local textures without high quality geometry. In this section, we review the literature for appearance rendering with a generative model and its application to synthesizing pose-guided human animation.

Generative Human Pose Transfer. Pose transfer refers to the problem of synthesizing human images with a novel user-defined pose. The conditioning pose is often captured by 2D keypoints [179, 180, 181, 182, 183, 184] or parametric mesh [3, 185, 186, 18]. Many recent works also use Densepose [187] which is the projection of SMPL model with UV parameterization in the image coordinates, as conditioning input. This enables direct warping of pixels of the input image to the spatial locations at the output with

target pose [185, 186, 18]. While the aforementioned methods produce photo-realistic results within the same dataset, they often exhibit serious artifacts on in-the-wild scenes, such as pixel blending around the boundaries between the different garment types.

To address these limitations, some recent methods use garment segmentation map, *i.e.,* a label image where each pixel belongs to a semantic category such as clothing, face, and arm, as input to a neural network [188, 189, 190, 191]. [192] condition garment type, whereas [193] handles each garment parts in different transformation layers to preserve the clothing style in the generated image. However, these works still do not generalize to new appearances and unseen scenes.

Some new methods explicitly handle appearance in the occluded areas by matching their style to the visible regions. [194] transforms the features of the input image to a target body pose with bidirectional implicit affine transformation. [195, 196] learn pixel-wise appearance flow in an unsupervised way based on the photometric consistency. [195] establishes direct supervision by fitting a body model to the images. However, the predicted warping fields is often unstructured, resulting in artifacts such as shape distortion.

Pose-Guided Video Generation. Since the methods for pose transfer are designed to output a single image, their application to a sequence of poses to perform pose guided video generation can exhibit temporal inconsistency. To mitigate this problem, many methods enforce explicit temporal constraints in their algorithm. [26] predicts the person image in two consecutive frames. [197] conditions the temporally coherent semantics on a generative adverserial network. Recent video generation approaches have leveraged the optical flow prediction [198], local affine transformation [199], body parts transformation [200], and future frame prediction [201, 202] to enforce the temporal smoothness. [203] learns to predict a dynamic texture map that allows rendering physical effects, *e.g.,* pose-dependent clothing deformation, to enhance the visual realism on the generated person. Unfortunately, the above methods are either person-specific or requiring the fine-tuning on unseen subjects for the best performance. While few-shot video generation [204] addressed this generalization problem, it still requires fine-tuning on the testing scene to achieve full performance. In contrast, our method works with a single conditioning image in the wild and performs pose guided video synthesis without any fine-tuning.

Neural Rendering. As human appearances are modulated by their poses, it is possible to generate the high fidelity appearance by using a parametric 3D body model, e.g., deformable template models [17]. For example, SMPLpix [205] learned a constant appearance of a person by mapping from per-vertex RGB colors defined on the SMPL body to synthesized images. Textured neural avatars [206] learned a person-specific texture map by projecting the image features to a body surface coordinate (invariant to poses) to model human appearances. These approaches, however, are limited to statics, i.e., the generated appearance is completely blind to pose and motion.

To model pose-dependent appearances, Liu et al. [207] implicitly learned the texture variation over poses, which allowed them to refine the initial appearance obtained from texture map through a template model. On the other hand, Raj et al. [208] explicitly learned pose-dependent neural textures. To further enhance the quality of rendering, person-specific template models [209] were used by incorporating additional meshes representing the garments [210]. However, none of these approaches are capable of modeling the time-varying secondary motion. Habermann et al. [62] utilized a motion cue to model the motion-dependent appearances while requiring a pre-learned person-specific 3D template model. Zhang et al. [211] proposed a neural rendering approach to synthesize the dynamic appearance of loose garments assuming a coarse 3D garment proxy is provided. In contrast, our method uses the 3D body prior to model the dynamic appearance of both tight and loose garments.

To model pose-dependent appearances, Liu et al. [207] implicitly learned the texture variation over poses, which allowed them to refine the initial appearance obtained from texture map through a template model. On the other hand, Raj et al. [208] explicitly learned pose-dependent neural textures. To further enhance the quality of rendering, person-specific template models [209] were used by incorporating additional meshes representing the garments [210]. However, none of these approaches are capable of modeling the time-varying secondary motion. Habermann et al. [62] utilized a motion cue to model the motion-dependent appearances while requiring a pre-learned person-specific 3D template model. Zhang et al. [211] proposed a neural rendering approach to synthesize the dynamic appearance of loose garments assuming a coarse 3D garment proxy is provided. In contrast, our method uses the 3D body prior to model the dynamic appearance of both tight and loose garments.

The requirement of the parametric model can be relaxed by leveraging flexible neural rendering fields. For instance, neural volumetric representations [212] have been used to model general dynamic scenes [213, 214, 215, 216, 217] and humans [218, 219] using deformation fields. Nonetheless, the range of generated motion is still limited. Recent methods learned the appearance of a person in the canonical space of the coarse 3D body template and employ skinning and volume rendering approaches to synthesize images [220, 221, 222]. Liu et al. [223] extended such approaches by introducing pose-dependent texture maps to model pose-dependent appearance. Modeling time-varying appearance induced by the secondary motion with such volumetric approaches is still the uncharted area of study.

Generative Human Image Synthesis. Generative adversarial learning enforces a generator to synthesize photorealistic images that are almost indistinguishable from the real images. For example, image-to-image translation can synthesize pose-conditioned appearance of a person by using various pose representations such as 2D keypoints [179, 182, 183, 181, 199, 224, 196], semantic labels [188, 189, 190, 193, 197, 195, 225], or dense surface coordinate parametrizations [186, 226, 18, 227]. Despite remarkable fidelity, these were built upon the 2D synthesis of static images at every frame, in general, failing to generate physically plausible secondary motion. To address this challenge, several works have utilized temporal cues either in training time [26] to enable temporally smooth results or as input signals [198, 204] to model motion-dependent appearances. Kappel et al. [28] modeled the pose-dependent appearances of loose garments conditioned on 2D keypoints by learning temporal coherence using a recurrent network. However, due to the nature of 2D pose representations, the physicality of the generated motion is limited, e.g., our experiments show that the method works well mostly for planar motions but is limited in expressing 3D rotations. Wang et al. [4] is the closest to our work, which maps a sequence of dense surface parameterization to motion features that are used to synthesize dynamic appearances using StyleGAN [228]. In contrast, our method is built on a new 3D motion representation which shows superior discriminative power, consistently outperforming [4] in terms of generalizing to unseen poses as demonstrated in our experiments.

Part I

Learning to Reconstruct 3D Avatars from a Single Camera

Chapter 3

Multiview Human Visual Dataset

Humans possess a quintessential sensitivity to effortlessly read invisible internal states of others, *e.g.*., intent, emotion, and attention, through every nuance of their body expressions, including gaze, face, and gestures. It is impossible, therefore, to enable authentic social presence in a virtual space without conveying photorealistic models of such body expressions. This is, however, extremely challenging because it requires decoding complex physical interactions between texture, geometry, illumination, and viewpoint (*e.g.*., translucent skins, tiny wrinkles, and reflective fabric) from an image of a subject.

Recently, pose- and view-specific models by making use of a copious capacity of neural encoding [60, 229] substantially extend the expressibility of existing linear models [127]. So far, these models have been constructed by a sequence of the detailed scans of a target subject using dedicated camera infrastructure (*e.g.*., multi-camera systems [121, 230, 231]), *i.e.*, they are subject-specific which is not generalizable to other subjects. Looking ahead, we would expect a new versatile model that is applicable to the general appearance of assorted people by eliminating the requirement of the massive scans for every target subject.

Among many factors, what are the core resources to build such a generalizable model? We argue that the data that can span an extensive range of appearances from numerous shapes and identities are prerequisites. To validate our conjecture, we present a new dataset of human body expressions called *HUMBI* (HUman Multiview Behavioral Imaging) that pushes to two extremes: views and subjects. As shown in Figure 3.1,

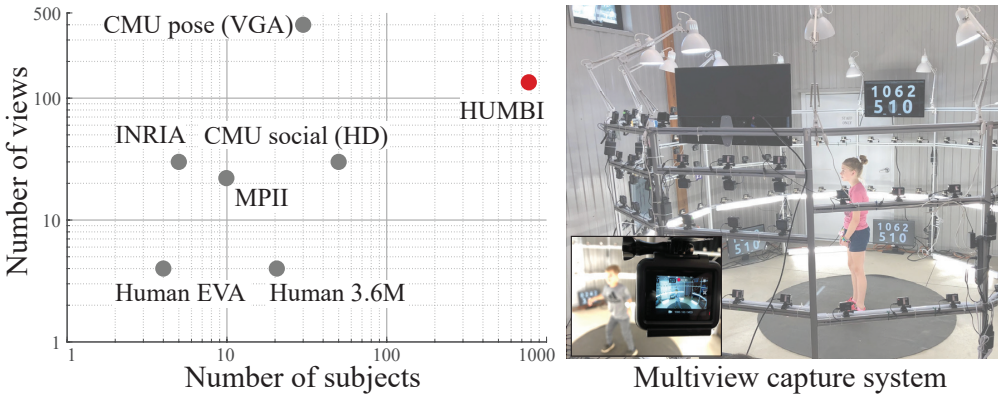


Figure 3.1. We present HUMBI that pushes towards two extremes: views and subjects. Comparing to existing datasets such as CMU Panoptic Studio [7, 8], MPII [9, 10], and INRIA [11], HUMBI presents the unprecedented scale visual data measured by 107 HD cameras that can be used to learn the detailed appearance and geometry of five elementary human body expressions for 772 distinctive subjects.

HUMBI is composed of 772 distinctive subjects with natural clothing across diverse age, gender, ethnicity, and style captured by 107 HD synchronized cameras (68 cameras facing at frontal body). This poses unprecedented diversity of visual data that are ideal for modeling generalizable geometry and appearance, which is not presented in existing datasets including CMU [7, 8] and MPII [9, 10] as shown in Figure 3.1.

Owing to these properties, HUMBI is an ideal dataset to evaluate the ability of modeling human appearance and geometry as shown in Figure 3.2. To measure such ability, we formulate a novel benchmark challenge on a pose-guided appearance rendering task: given a single view image of a person, render the person appearance from other views and poses. HUMBI offers the ground truth of this challenging task where the performance of the approaches can be precisely characterized. We validate the feasibility of the benchmark challenge using the state-of-the-art rendering methods [179, 181, 184, 196].

3.1 Multi-Camera Imaging System

We design a re-configurable multi-camera system that was deployed in public events including Minnesota State Fair and James Ford Bell Museum of Natural History at the University of Minnesota.

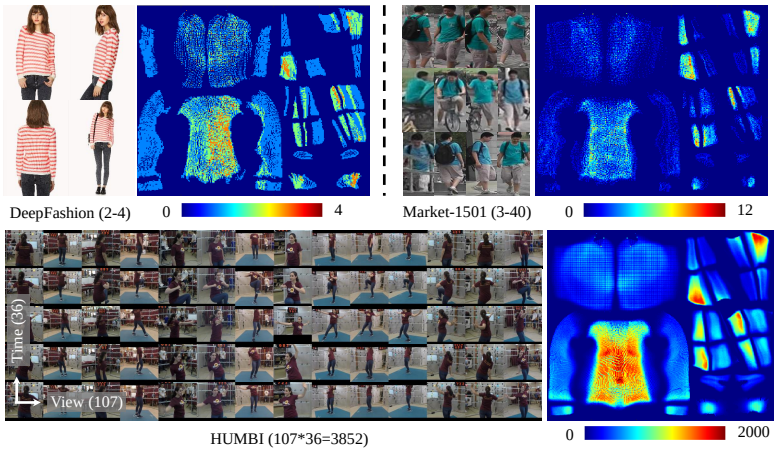


Figure 3.2. The existing datasets (Deepfashion [1] and Market-1501 [12]) are designed for the task of person re-identification and fashion retrieval, which includes the images captured from limited viewpoints. On the other hand, HUMBI provides images captured from dense camera array, which is ideal to develop and evaluate a human rendering model. The body surface visibility for each dataset is visualized [13]. The colormap describes the number of cameras visible at each pixel.

Hardware The stage is made of a re-configurable dodecagon frame with 4.2 m diameter and 2.5 m height using T-slot structural framing (80/20 Inc.) where the baseline between adjacent cameras is approximately 10° (22 cm) as shown in Figure 3.3. The stage is encircled by 107 GoPro HD cameras (38 HERO 5 BLACK Edition and 69 HERO 3+ Silver Edition), one LED display for an instructional video, eight LED displays for video synchronization, and additional lightings. Among 107 cameras, 69 cameras are uniformly placed along the two levels of the dodecagon arc (0.8 m and 1.6 m) for body and clothing, and 38 cameras are place over the frontal hemisphere for face and gaze.

Instructional Performance Guidance To guide the movements of the participants, we create four instructional videos (∼2.5 minutes). Each video is composed of four sessions. (1) Gaze: subjects were asked to find and look at the requested number tag posted on the camera stage; (2) Face: subjects were asked to follow 20 distinctive dynamic facial expressions (*e.g.*., eye rolling, frowning, and jaw opening); (3) Hand: subjects were asked to follow a series of American sign languages (*e.g.*., counting one to ten, greeting, and daily used words); (4) Body and garment: subjects were asked to

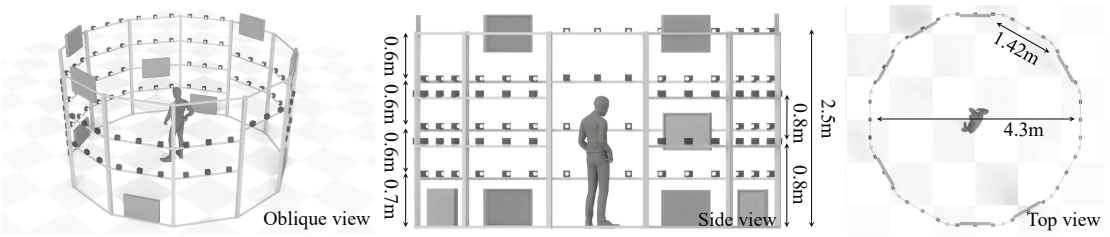


Figure 3.3. Re-configurable dodecagon design and its dimension of the multi-camera system.

follow range of motion, which allows them to move their full body and to follow slow and full speed dance performances curated by a professional choreographer.

Synchronization and Calibration We manually synchronize 107 cameras using LED displays. The maximum synchronization error is up to 15 ms. We use the COLMAP [76] software to calibrate camera intrinsics and extrinsics parameters and upgrade the extrinsic parameters to a metric space: the scale is corrected using physical distance between cameras, the origin is translated to the center of the stage, and the orientation is mapped such that its y-axis is aligned with the surface normal of the ground plane.

3.2 HUMBI

HUMBI is composed of 772 distinctive subjects, where each subject includes five elementary body expressions: gaze, face, hand, body, and garment. Notable subject statistics includes: evenly distributed gender (50.7% female; 49.3% male); a wide range of age groups (26% of teenagers, 29% of 20s, and 11% of 30s); diverse skin colors (black, dark brown, light brown, and white); various styles of clothing (dress, short-/long-sleeve t-shirt, jacket, hat, and short-/long-pants) as shown in Figure 3.4.

Notation We denote our representation of human body expressions as follows:

- 3D keypoints: $\mathcal{K}$.
- 3D vertices: $\mathbf{V}$.
- 3D occupancy map: $\mathcal{O} : R^3 \rightarrow \{0, 1\}$ that takes as input 3D voxel coordinate and outputs binary occupancy.

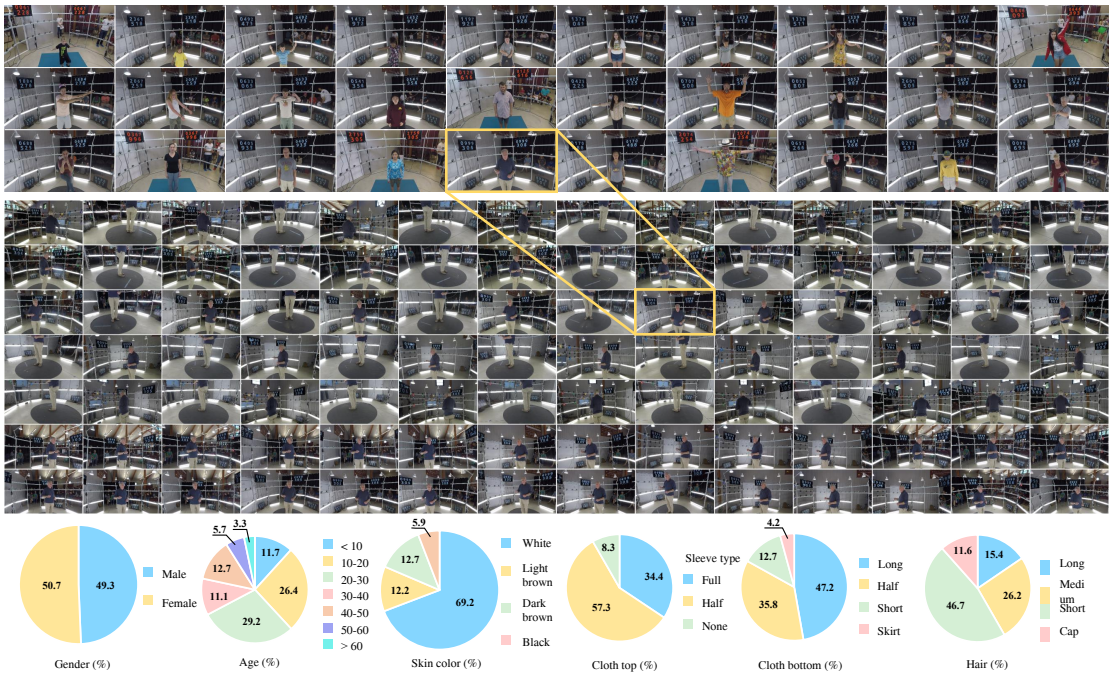


Figure 3.4. (Top and bottom) HUMBI includes 772 distinctive subjects across gender, ethnicity, age, clothing style, and physical condition, which generates diverse appearance of human expressions. (Middle) For each subject, 107 HD cameras capture her/his expressions including gaze, face, hand, body, and garment.

- Appearance map: $\mathcal{A} : R^2 \rightarrow [0,1]^3$ that takes as input atlas coordinate (UV) and outputs normalized RGB values.

3.2.1 Body

HUMBI Body contains 26M images (315 frames $\times$ 107 views per subject). We present body geometry using a 3D linear blend shape model [17] with 4,129 vertices and 7,999 faces without hand and head parts:

$$\mathbf{V}(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(\bar{\mathbf{V}} + \mathcal{T}(\boldsymbol{\beta}, \boldsymbol{\theta}), \mathcal{K}(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathcal{W}), \tag{3.1}$$

where $\mathbf{V} \in \mathbb{R}^{3\times D}$ is the vertices of the posed 3D body ($D = 4,129$), and W is the skinning function [17] that takes the mean body shape in the rest pose $\bar{\mathbf{V}}$, pose and

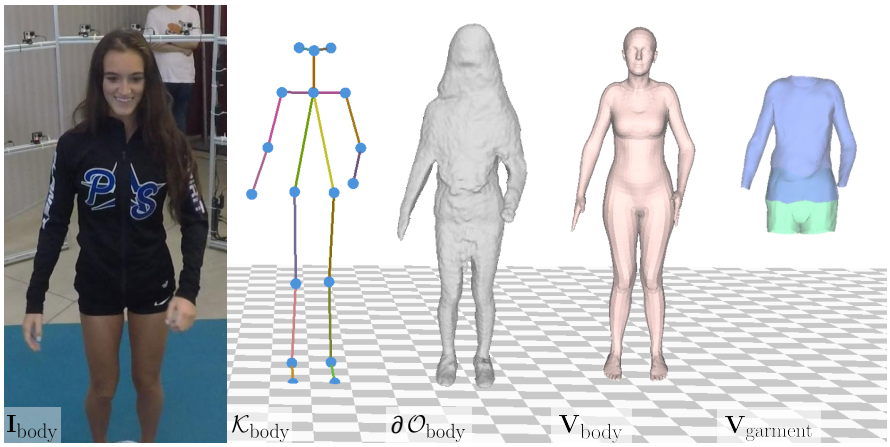


Figure 3.5. Body and clothing reconstruction results.

shape blending shapes $\mathcal{T}$, 3D keypoints $\mathcal{K}$, and blending weights $\mathcal{W}$. This skinning function is parameterized by the shape $\boldsymbol{\beta} \in R^{10}$ and pose coefficients $\boldsymbol{\theta} \in R^{24\times 3}$ with axis-angle representation, where $\boldsymbol{\theta}_1 \in \mathbb{R}^{1\times 3}$ represents the root orientation and others the relative angles with respect to the root joint.

We reconstruct the body model by minimizing the following cost:

$$E(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{t}, \boldsymbol{s}) = E^p + \lambda^s E^s + \lambda^r E^r, \tag{3.2}$$

where λ^s and λ^r control the importance of each measurement, and $\boldsymbol{t} \in R^3$ and $\boldsymbol{s} \in R^+$ represent the global translation and scale, respectively.

Given the correspondences between the reconstructed keypoints (*i.e.*, $\mathcal{K}_{\text{body}}$ in Figure 3.5) and the body mesh, we recover the posed body model by minimizing the keypoints error:

$$E^p(\boldsymbol{\theta},\ \boldsymbol{\beta}, \boldsymbol{t}, \boldsymbol{s}) = \sum_i \|\mathcal{K}_i - \mathbf{V}_i\|^2 . \tag{3.3}$$

We recover the shape of the body model by aligning the model to the surface of the 3D reconstruction (*i.e.*, $\partial\mathcal{O}_{\text{body}}$ in Figure 3.5). We use Chamfer distance to measure their alignment:

$$E^s(\boldsymbol{\beta},\ \boldsymbol{\theta},\ \boldsymbol{t}, \boldsymbol{s}) = d_{\text{chamfer}}(\partial\mathcal{O}, \mathbf{V}), \tag{3.4}$$

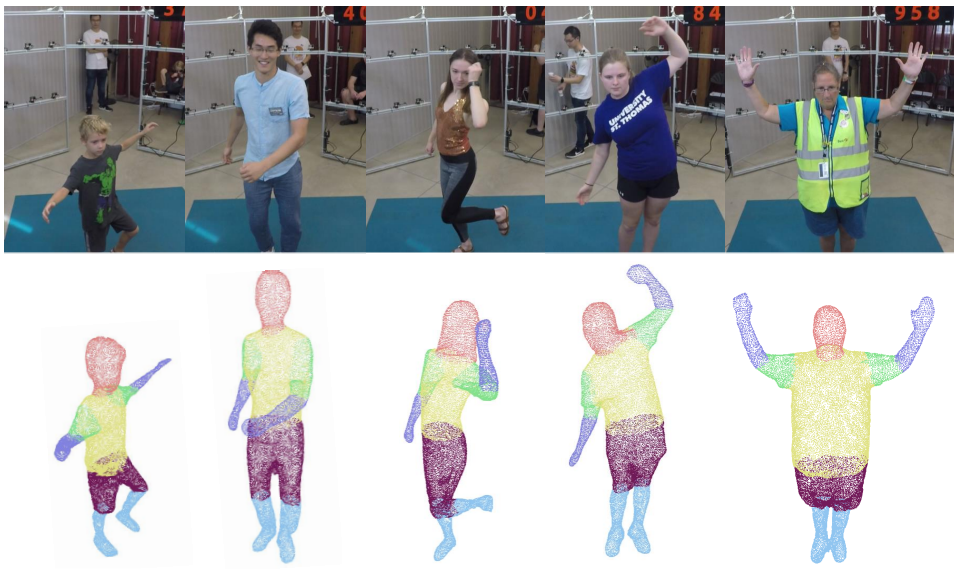


Figure 3.6. We reconstruct the body occupancy map and its outer surface using shape-from-silhouette and associate the point cloud with body semantics (head, body, arms, and legs).

where d_{chamfer} measures Chamfer distance between two sets of point clouds, $\partial\mathcal{O} \in R^{3\times N}$ is a set of the 3D points on the outer surface of the occupancy map, and N is the number of the 3D points. We use Shape-from-silhouette[1] [232] to reconstruct the occupancy map $\mathcal{O}$ with human body parts segmentation [233]. As a by-product, the semantics (*i.e.*, head, torso, upper arm, lower arm, upper leg, and lower leg) can be labeled at each location in the occupancy map by associating with the projected body label [66] as shown in Figure 3.6.

E^r penalizes the difference between the estimated shape $\boldsymbol{\beta}$ and the subject-specific mean shape $\boldsymbol{\beta}^{\text{prior}}$ as follows:

$$E^r(\boldsymbol{\beta}; \boldsymbol{\beta}^{\text{prior}}) = \left\|\boldsymbol{\beta} - \boldsymbol{\beta}^{\text{prior}}\right\|^2. \tag{3.5}$$

This prevents unrealistic shape fitting due to the estimation noise/error, *e.g.*., erroneous surface reconstruction of the body parts due to the occlusion by hands. To obtain the shape prior $\boldsymbol{\beta}^{\text{prior}}$, we solve the Eq. (3.2) without E^r and take the median $\boldsymbol{\beta}$ over time.

[1] MultiView stereo reconstruction [76] is complementary to the shape-from-silhouette.

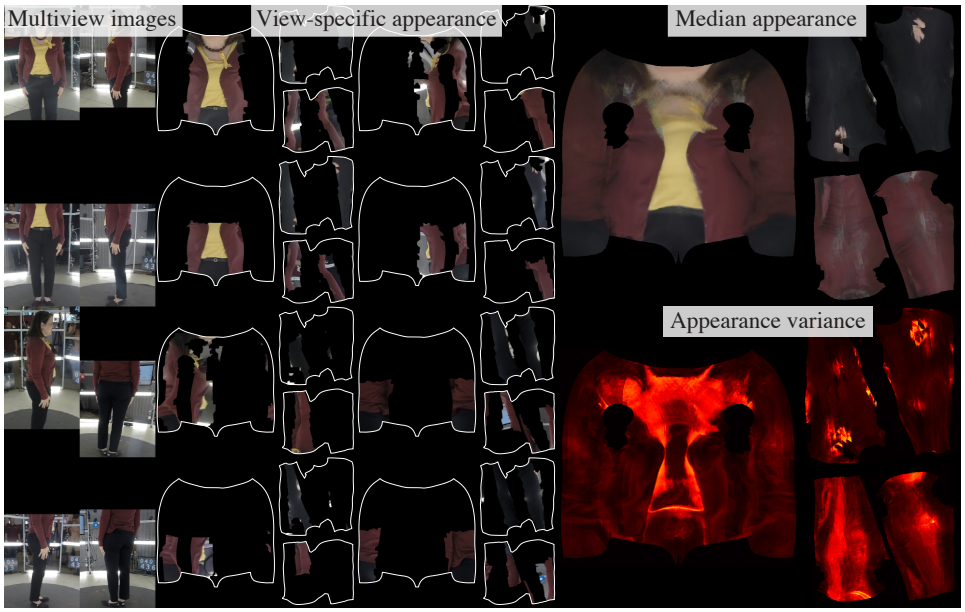


Figure 3.7. The view-specific body appearance rendered from multiview images with its median and variance.

Given the reconstructed body mesh model, we construct a view-specific appearance map $\mathcal{A}$ (1024 $\times$ 1024 pixels) by projecting the pixels in an image onto the canonical atlas coordinate. Figure 3.7 illustrates view-specific appearance across views with its median and variance of appearance. The variance map shows that the appearance is dependent on viewpoints.

To reconstruct 3D Body Keypoint ($\mathcal{K}_{\text{body}} \in R^{3\times 25}$), we detect 2D keypoints of body (including feet) [25] given a set of synchronized and undistorted multiview images. Using these keypoints, we triangulate 3D keypoints with RANSAC [234] followed by the non-linear refinement by minimizing reprojection error [235][2].

3.2.2 Garment

Similar to HUMBI Body, HUMBI Garment includes 26M images. Given the body reconstruction, we represent the garment geometry using an in-house garment mesh

[2]When multiple persons are detected, we use a geometric verification to identify each subject.

model[3] $\mathbf{V} \in R^{3\times D}$, where D is the number of 3D points. Unlike parametric models used in face, hand, and body, there exists no shape model that can express diverse topology, style, and type of garments. Instead, we represent the dynamic garment shape with per-vertex 3D warping fields [236] that map the garment mesh vertices at the rest pose $\bar{\mathbf{V}}$ to the deformed garment:

$$\mathbf{V}_i = \mathbf{R}_i \bar{\mathbf{V}}_i + \mathbf{t}_i, \tag{3.6}$$

where $(\mathbf{R}_i, \mathbf{t}_i) \in SE(3)$ is the $6D$ transformation. We optimize this warping field by minimizing the following objective:

$$E(\mathbf{R}, \mathbf{t}) = E^c + \lambda^o E^o + \lambda^r E^r, \tag{3.7}$$

where λ^o and λ^r are weight parameters.

We predefine a set of fiducial correspondences between the garment and body meshes, which are the control points to deform the garment mesh. E^c measures the correspondence error:

$$E^c(\mathbf{R}, \mathbf{t}) = \sum_i \|\widehat{\mathbf{V}}_i^b - \widehat{\mathbf{V}}_i^g\|^2, \tag{3.8}$$

where $\widehat{\mathbf{V}}^g$ and $\widehat{\mathbf{V}}^b$ are the corresponding vertices between the garment and body model, respectively.

E^o measures Chamfer distance to align the garment mesh model with the 3D points on the outer surface of the occupancy map $\partial\mathcal{O} \in R^{3\times N}$ where N is the number of the correspondences:

$$E^o(\mathbf{R}, \mathbf{t}) = d_{\text{chamfer}}(\partial\mathcal{O}, \mathbf{V}). \tag{3.9}$$

E^r is a spatial regularization based on Laplacian mesh deformation [237] that enforces as-rigid-as-possible deformation by penalizing a non-smooth and non-rigid vertex with respect to its neighboring vertices:

$$E^r(\mathbf{R}, \mathbf{t}) = \nabla^2 \mathbf{V}. \tag{3.10}$$

[3] A similar approach was used to reconstruct garment from 4D scanner [10].

Training / Testing	H36M	MI3D	HUMBI	H36M +HUMBI	MI3D +HUMBI
H36M	0.562	0.362	0.434	0.551	0.437
MI3D	0.317	0.377	0.354	0.375	0.425
HUMBI	0.248	0.267	0.409	0.372	0.377
Average	0.376	0.335	**0.399**	**0.433**	**0.413**

Table 3.1. The cross-data evaluation of 3D body keypoint prediction. AUC of PCK is used for a metric over an error range of 0-150 mm.

We use three garment topologies, *i.e.*, tops: sleeveless shirts (3,763 vertices and 7,261 faces), T-shirts (6,533 vertices, 13,074 faces), and long-sleeve shirts (8,269 vertices and 16,374 faces), and bottoms: short (3,975 vertices and 7,842 faces), medium (5,872 vertices and 11,618 faces), and long pants (11,238 vertices and 22,342 meshes), which are manually matched to each subject.

3.3 Validation

We evaluate HUMBI in terms of generalizability and accuracy. For generalizability, we conduct the cross-data evaluation on the tasks of single view human reconstruction, *e.g.*., monocular 3D face mesh prediction. For accuracy, we measure the silhouette similarity between the human annotation and the reprojection of the reconstructed 3D model.

3.3.1 Body

Baseline Dataset We use four baseline datasets: (1) Human3.6M [14] contains numerous 3D human poses of 11 actors/actresses measured by motion capture system with corresponding images from 4 cameras. (2) MPI-INF-3DHP [238] is a 3D human pose estimation dataset which is composed of images with 2D and 3D pose labels captured in both indoor and outdoor scenes. We use its test set containing 2,929 valid images from 6 subjects. (3) UP-3D [82] is a 3D body mesh dataset including 9K images with 3D body meshes. We use Human3.6M and MPI-INF-3DHP for body pose evaluation and UP-3D for body mesh evaluation.

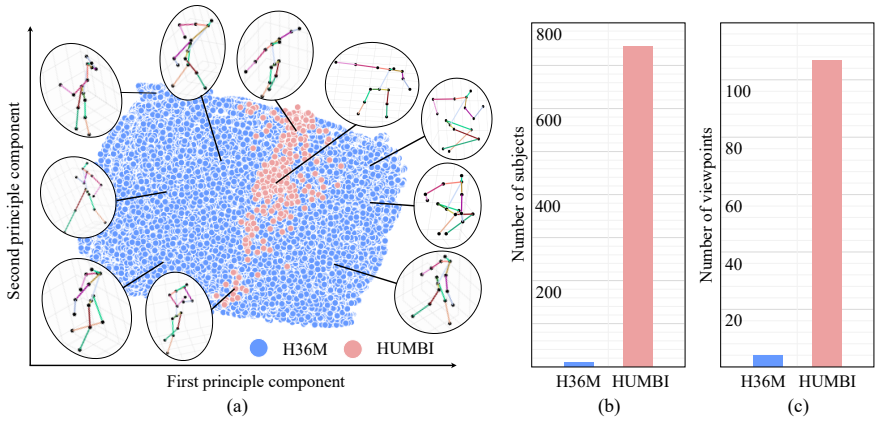


Figure 3.8. The comparison of the dataset distribution between Human3.6M (H36M) [14] and HUMBI Body. (a) The distribution of the 3D poses *per subject* in each dataset. We visualize the first and second principal components of the normalized 3D poses where each joint is represented by unit vectors. (b) The number of subjects in each dataset. (c) The number of camera viewpoints in each dataset.

Monocular 3D Body Pose Prediction We conduct a cross-data evaluation for the task of estimating 3D human pose from a single view image. We use the 3D body pose detector [239] as a base network. We train the model on each dataset alone as well as a mix of HUMBI Body and each of the other two datasets. We evaluate the resulting models on each of those 3 datasets. We use 2D landmark labels from MPII dataset [240] as a weak supervision similar to the training scheme of [239]. The results are summarized in Table 3.1. We use the area under PCK curve (AUC) in an error range of 0-150 mm as the metric. HUMBI shows superior performance on predicting 3D body pose comparing to Human3.6M and MPI-INF-3DHP with a margin of 0.023 and 0.064 AUC. Moreover, HUMBI is complementary to each dataset, *i.e.*, the performance of the model trained by another dataset is increased (by a margin of 0.057 and 0.078 AUC, respectively. We further demonstrate the complementary nature of HUMBI by comparing the pose distribution of HUMBI and Human3.6M (H36M) [14] as shown in Figure 3.8. H36M provides assorted 3D poses per subject, e.g., HUMBI does not include sitting poses, while HUMBI provides the appearance of diverse subjects seen from a number of viewpoints.

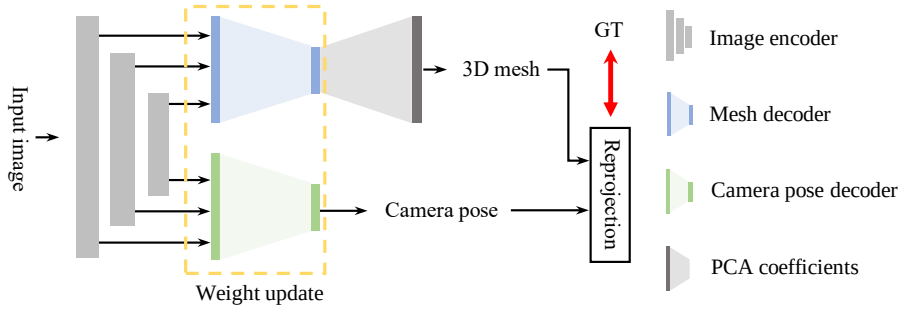


Figure 3.9. We use a vanilla network [15] design to evaluate the strength of the datasets. This network takes as input an image and outputs the parameters of the 3D mesh and camera pose. The network is made of the pre-trained image encoder [16] that extracts image features and two decoders that predict the latent mesh parameters and camera pose where we train these decoders from scratch by minimizing the reprojection error. From the predicted model parameters, we reconstruct the 3D body shape using the PCA coefficients of the body model (SMPL [17]) for body.

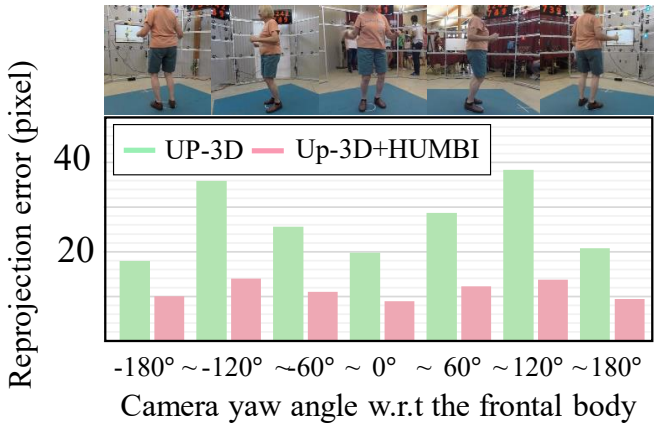


Figure 3.10. We measure the viewpoint dependency of body mesh reconstruction models. Combining with HUMBI enforces learning a representation agnostic to viewpoints.

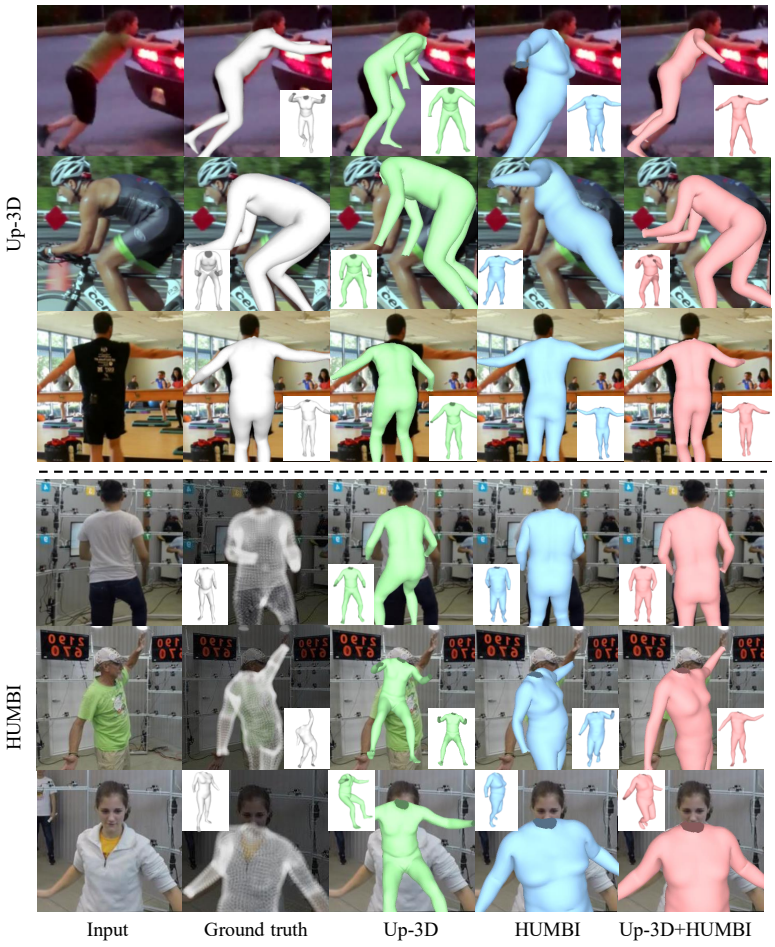


Figure 3.11. The qualitative results of the monocular 3D body prediction network trained on different data combination. The column and row represent the type of training and testing data, respectively.

Training / Testing	UP-3D	HUMBI	UP-3D+HUMBI
UP-3D	22.7±18.6	49.4±0.09	**18.4±13.8**
HUMBI	26.0±19.7	14.5±6.6	**12.5±8.4**

Table 3.2. The mean error of 3D body mesh prediction for cross-data evaluation (unit: pixel).

Monocular 3D Body Mesh Prediction We evaluate HUMBI Body by predicting a 3D body mesh using a vanilla mesh reconstruction network [15] as described in Figure 3.9. The network is composed of an image encoder that extracts image features and two decoders: mesh decoder that predicts the shape coefficients and camera pose decoder that estimates the camera extrinsic parameters. We use the pre-trained image encoder [15] while two decoders are trained from scratch. The vanilla network model trained on (1) HUMBI, (2) UP-3D, and (3) HUMBI+UP-3D. The mesh decoder generates SMPL parameters, and the camera pose decoder estimate the camera extrinsic parameters. We train these two decoders by minimizing the reprojection error with the multiview annotations. The cross-data evaluation is summarized in Table 3.2 and the associated qualitative comparisons are shown in Figure 3.11. We observe that the network trained with HUMBI shows weak performance due to the lack of diversity in poses. However, the performance of the model trained by the combined datasets (*i.e.*, HUMBI+UP-3D) shows an increase of 2 pixels from the model trained by HUMBI alone and 4.3 pixels from Up-3D, indicating that HUMBI is highly complementary to the other datasets. Further, Figure 3.10 shows that HUMBI is effective to alleviate the viewpoint bias of the existing dataset.

3.3.2 Garment

Unlike other body expressions, we validate HUMBI Garment with the geometric accuracy because there exists no public dataset that provides a 3D garment model and the associated multiview images[4]. To measure the geometric accuracy, we use two metrics: Intersection over Union (IoU) and Chamfer distance between the ground truth mask and

[4]The existing real datasets are either 3D [46] or 2D [1]

Type \ Style	Short (IoU/Chamf)	Half (IoU/Chamf)	Long (IoU/Chamf)
Top	0.73 / 10.9	0.90 / 5.10	0.85 / 7.38
Bottom	0.86 / 6.38	0.83 / 9.07	0.87 / 6.27

Table 3.3. The summary of the garment reconstruction accuracy. We measure the accuracy with the Intersection over Union (IoU) and Chamfer distance (unit: pixel) between the ground truth and the reprojection of the 3D garment.

the one from the 2D reprojection of the reconstructed 3D garment mesh. We manually segment the ground truth region using an interactive segmentation tool [241] considering the occlusion. We subsampled five subjects for each garment model (sleeveless shirts, half-sleeve shirts, long-sleeve shirts, short pants, half pants, and long pants as introduced in Section 3.2.2) and report the mean accuracy in Table 3.3. Overall, the geometric accuracy of our clothing reconstruction shows more than 0.84 overlap ratio with the ground truth mask and less than 7.51 pixel distance from the annotated garment boundary on average. In Figure 3.12, we visualize the silhouette of the reconstructed 3D garment overlayed with the ground truth. In addition, we provide the evaluation on the view-dependency of the garment reconstruction. For this, we pick a half-sleeve shirts and half pants models as a representative garment of top and bottom and measure the accuracy based on the Chamfer distance from each camera view that has different angle with respect to the most frontal camera. On average shown in Figure 3.13, the silhouette error seen from the side view (11 pixels) is higher than the frontal (7.5 pixels) and rear views (8 pixels).

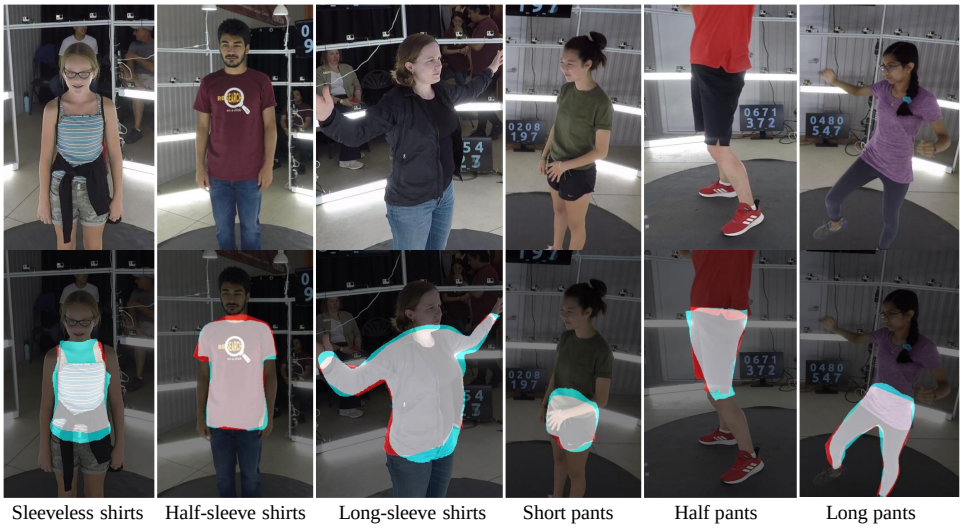


Figure 3.12. Silhouette of the reconstructed 3D garments overlayed with ground truth. The model is visualized with the blue, the ground truth with red, and the overlap with white.

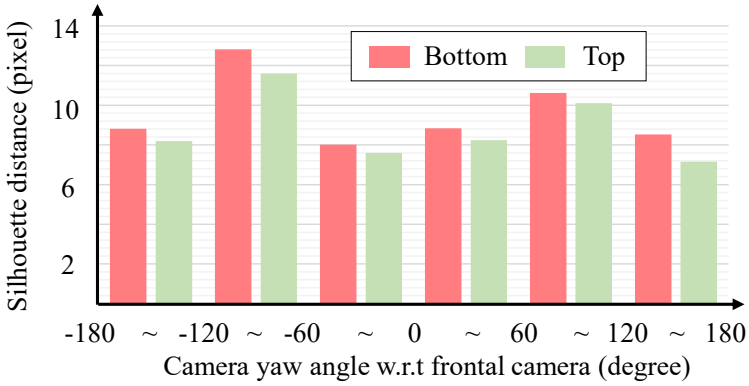


Figure 3.13. Garment silhouette error.

3.3.3 Benchmark Challenge

HUMBI provides multiview images of many subjects with diverse poses, which offers a unique opportunity to evaluate a task of human appearance rendering. We formulate a new benchmark challenge of rendering that can facilitate photo-realistic human rendering research.

Task Definition Given an image of a person and a target pose, render the appearance of the person that agrees with the target pose.

Benchmark Dataset We randomly select the pair of the reference and target poses across the time and views. In our experiments, 101K pairs of the views are selected from 100 subjects for training, and 15,923 from 40 subjects for testing. For each view, we use the person image with 256×256 pixel resolution after cropping and resizing based on the projected xy-coordinate of 3D keypoints.

Baselines We evaluate the following six state-of-the-art approaches. **PG** [179] synthesizes a person image with two generators in coarse-to-fine manner: **PG-1** and **PG-2** indicates coarse and fine syntheses. **C2GAN** [184] uses the cycle consistency on body keypoints and multiview images. **PPA** [181] integrates the pose attention module into a generative network to progressively refine the image, and **SGAN** [242] selectively combines multi-channel attention map that enhances the quality of the generated images. For **PPA**, we use the image with 256×176 pixel resolution. **GFLA** [196] estimates a global flow field to transform the local attention features. **NHRR** [18] warps the pixels from the input image to the target image based on the dense correspondences from a parametric body model [17]. We train these networks using the training parameters suggested by the authors.

Metric We measure the quality of the generated images using Learned Perceptual Image Patch Similarity (LPIPS) [63] and Frechet Inception Distance (FID) [243]. LPIPS measures the distance between the generated images and ground truth in a feature space, e.g., VGG features. FID measures the realism of the generated images by computing Wassertein-2 distance between the distributions of the generated images and ground truth. To eliminate the influence of background, we mask out the background region by incorporating segmentation [244] to form Mask-LPIPS and Mask-FID.

Analysis Table 3.4 summarizes the quantitative evaluations on HUMBI benchmark

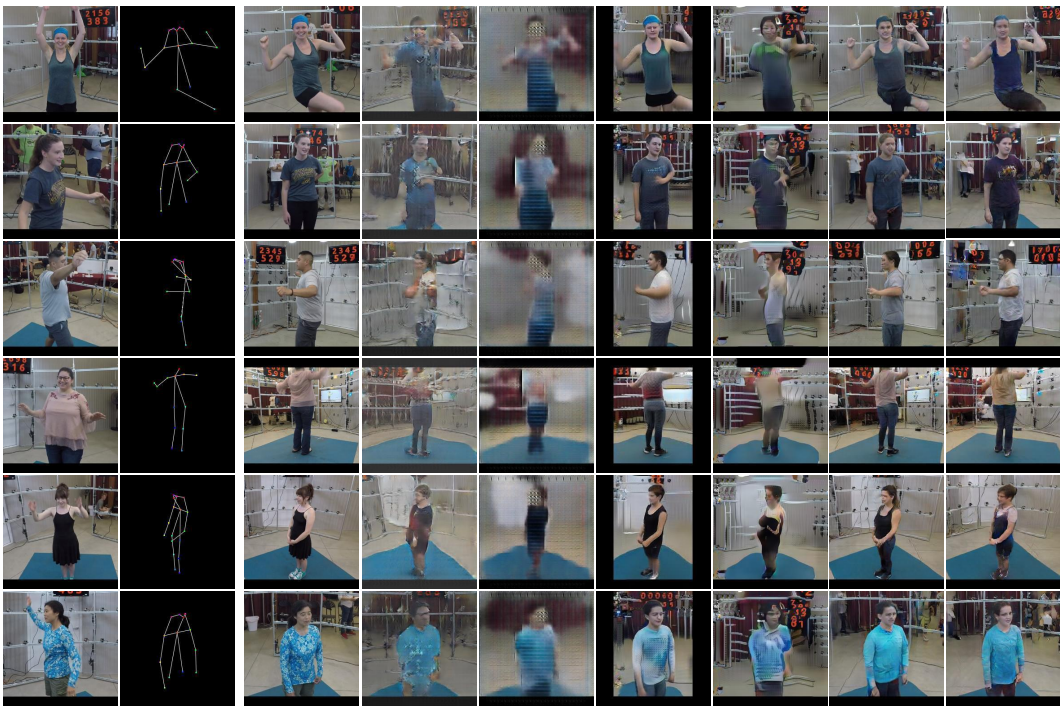


Figure 3.14. The qualitative comparison of pose-guided person image generation from each method. For NHRR [18], the densepose detection [13] is used as a conditioning target pose.

dataset. GFLA outperforms other methods. It can effectively model the following view-dependent properties by learning our multiview dataset: 1) The network generates the realistic background scene which is dependent on the camera viewpoint as shown in Figure 3.14. 2) The network models the view-dependent lighting, that can be verified in Figure 3.14-(third row), e.g., the color of T-shirt is bluish from the source view, while it is grayish from the target view. This indicates that the network can implicitly model the camera viewpoint from the target body pose and decode such view-dependant properties on the generated images. The comparison of FID with Mask-FID shows that the performance of PPA is significantly improved with the human mask, highlighting that it focuses only on the person region NHRR shows the best performance on the Mask-LPIPS metric, i.e., the rendered human images are perceptually close to real.

It is also worth to note several limitations observed from these state-of-the-art approaches. The person specific visual features, e.g., color and shape of the face and hair,

Metric / Baseline	LPIPS	Mask-LPIPS	FID	Mask-FID
PG-1 [179]	0.680±0.076	0.154±0.048	290.45	139.44
PG-2 [179]	0.465±0.064	0.141±0.043	105.56	80.64
C2GAN [184]	0.641±0.066	0.166±0.048	241.44	148.55
PPA [181]	0.537±0.043	0.137±0.049	114.08	19.95
SGAN [242]	0.525±0.044	0.164±0.051	111.58	34.91
GFLA [196]	**0.328±0.090**	0.122±0.042	**14.25**	**13.49**
NHRR [18]	0.356±0.093	**0.115±0.038**	19.21	17.49

Table 3.4. The quantitative evaluation of pose-guided person image generation. The lower score shows the better results.

in the generated images are not photo-realistically rendered. Transferring a variety of clothing style is more challenging. For example, the dress in the reference image is transferred to long pants from the generated image as shown in Figure 3.14-(fifth row), and a lacy shirt is converted to a tight T-shirt in Figure 3.14-(fourth row). They also fail to transfer the clothing textures in a semantically meaningful way, e.g., the flower patterns on the clothing from the reference image in Figure 3.14-(sixth row) is not preserved in the generated ones. Likewise, human rendering from a single image of diverse subjects is still far behind the metric-level accuracy, i.e., the rendering does not match to the ground truth image at each pixel location, and there exists substantial rooms to improve. This will encourage future research to push the boundary of photorealistic human rendering with tera-scale multiview imaging dataset.

3.3.4 Summary

We present HUMBI dataset that is designed to facilitate high resolution pose- and view-specific appearance of human body expressions. Five elementary body expressions (gaze, face, hand, body, and garment) are captured by a dense camera array composed of 107 synchronized cameras. The dataset includes diverse activities of 772 distinctive subjects across gender, ethnicity, age, and physical condition. We use a 3D mesh model to represent the expressions where the view-dependent appearance is coordinated by its canonical atlas. Our evaluation shows that HUMBI outperforms existing datasets as

modeling nearly exhaustive views and can be complementary to such datasets.

The main properties of HUMBI are summarized below. (1) Complete: it captures the images of total human appearance, including gaze, face, hand, foot, body, and garment to represent holistic body signals [245], *e.g.*., perceptual asynchrony between the face and hand movements. (2) Dense: 107 HD cameras create a dense light field that observes the minute body expressions with minimal self-occlusion. This dense light field allows us to model precise appearance as a function of view [60]. (3) Natural: the subjects are all voluntary participants (no actor/actress/student/researcher). Their activities are loosely guided by performance instructions, which generates natural body expressions. (4) Diverse: it captures 772 distinctive subjects with diverse clothing styles, skin colors, time-varying geometry of gaze/face/body/hand, and range of motion. (5) Fine: with multiview HD cameras, we reconstruct the high fidelity 3D model using 3D meshes, which allows representing view-specific appearance in its canonical atlas. (6) Effective: we show that vanilla convolutional neural networks (CNN) designed to learn view-invariant 3D pose geometry from HUMBI quantitatively outperform the counterpart models trained by existing datasets. More importantly, we show that it is *complementary* to such datasets, *i.e.*, the trained models can be substantially improved by combining with these datasets.

Chapter 4

3D Semantic Trajectory Reconstruction

A 3D trajectory representation [106, 107, 246, 109, 121] is a viable computational model that measures microscopic human behavior at high spatial resolution without prior scene assumptions. Unfortunately, the representation is lacking semantics, *i.e.*, it is important to know not only where a 3D point is but also what it means and how associated with other points. For instance, as shown in Figure 4.1, the trajectory of the basketball player's hand (semantics) is spatially and temporally related with that of the ball, which can describe their physical interactions. In this paper, we present a method to precisely assign the semantic label on dense 3D trajectory stream reconstructed by a large scale multi-camera system that emulates the 3D pixel continuum.

4.1 System Overview

Our system takes synchronized multiview image streams from a multi-camera system. We use the standard structure from motion pipeline [235, 247] to calibrate the camera and reconstruct trajectory stream in 3D as described in Section 4.4. The 3D reconstructed trajectories are used to infer their semantic labels by consolidating 2D recognition confidence in multiple view images: 3D semantic map is constructed using view-pooling (Section 4.3.1), and affinity between long range fragmented trajectories is measured by computing local transformation (Section 4.3.2). The system outputs

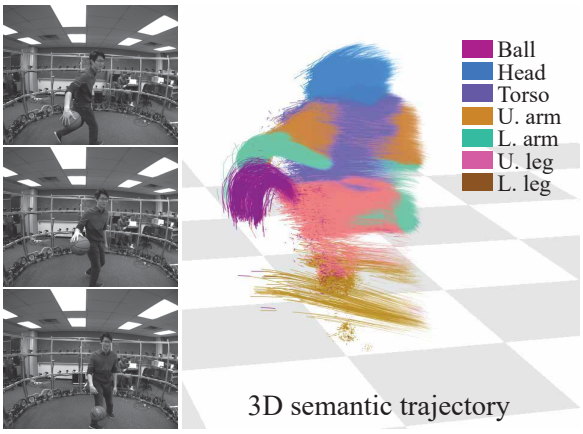


Figure 4.1. Given 3D dense reconstructed trajectories, we assign their semantic meaning using multiple view image streams. Each trajectory is associated with semantic labels such as body parts and objects (basketball). For illustrative purpose, the last 10 frames of trajectories are visualized.

the 3D dense semantic trajectories that consistently align with image visual semantic recognition.

4.2 Notation

We represent a fragmented trajectory with a time series of 3D points: $\mathcal{X} = \{\mathbf{X}_t \in R^3\}_{t=T_e}^{T_d}$ where $\mathbf{X}_t$ is the 3D point in the trajectory at the t time instant, and T_e and T_d are emerging and dissolving moments of the trajectory, respectively. We denote the probability of visibility as $V(\mathbf{X}_t, c) \in [0, 1]$ as shown in Figure 4.2 where $c \in \mathcal{C}$ is the camera index, and $\mathcal{C}$ is the camera index set, *i.e.*, $|\mathcal{C}|$ is the number of cameras.

The 3D point $\mathbf{X}_t$ is projected onto the visible c^{th} camera projection matrix, $\mathbf{P}_c = \mathbf{K}_c\mathbf{R}_c \begin{bmatrix} \mathbf{I}_3 & -\mathbf{C}_c \end{bmatrix} \in R^{3\times4}$ to form the 2D projection, $P(\mathbf{X}_t, c) \in R^2$ where $\mathbf{K}_c$ is the intrinsic parameter of the camera encoding focal length and principal points, and $\mathbf{R}_c \in SO(3)$ and $\mathbf{C}_c \in R^3$ are the extrinsic parameters (rotation and camera center), *i.e.*, $P(\mathbf{X}_t, c) = \begin{bmatrix} \mathbf{P}_c^1\widetilde{\mathbf{X}}_t/\mathbf{P}_c^3\widetilde{\mathbf{X}}_t & \mathbf{P}_c^2\widetilde{\mathbf{X}}_t/\mathbf{P}_c^3\widetilde{\mathbf{X}}_t \end{bmatrix}^{\mathsf{T}}$ where $\widetilde{\mathbf{X}}$ is the homogeneous representation of $\mathbf{X}$, and $\mathbf{P}_c^i$ indicates the i^{th} row of $\mathbf{P}_c$. We assume the camera extrinsic and intrinsic parameters are pre-calibrated and constant across time (no time index).

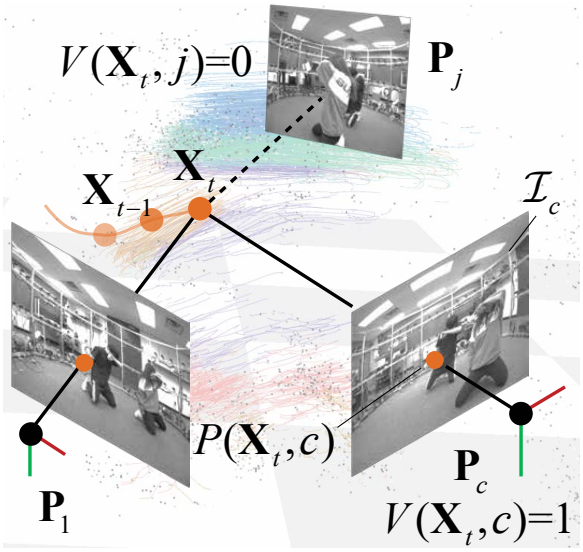


Figure 4.2. A 3D point $\mathbf{X}_t$ at the t time instant is observed by multiple cameras $\{\mathbf{P}_c\}_{c\in\mathcal{C}}$ where the point is fully visible to the c^{th} camera if $V(\mathbf{X}_t, c) = 1$, and zero otherwise. We denote the 2D projection of the 3D point onto the camera as $P(\mathbf{X}_t, c)$.

The c^{th} camera produces the image at the t time instant $\mathcal{I}_t^c$. Each pixel $\mathbf{x}$ is associated with the confidence of semantic labels, *i.e.*, $L_{2D}\left(\mathbf{x} \in R^2 | \mathcal{I}_c\right) \in [0,1]^N$ where N is the number of object classes[1]. For instance, L_{2D} can be approximated by the last layers of a convolutional neural network as shown in Figure 4.3. Our framework can build on general 2D recognition framework that can produce a confidence map while in this paper, we focus on two main pre-trained models: body semantic segmentation [19] and bounding box object recognition [20].

4.3 Semantic Trajectory Labeling

Given 3D reconstructed trajectories, we present a method to precisely infer their semantic labels. A key innovation is the *3D semantic map* that can encode the visual semantics of a 3D trajectory by consolidating the 2D recognition confidence across multiple view image streams. We integrate the 3D semantic map in conjunction with long

[1]The object classes include objects, body parts, and independent instances.

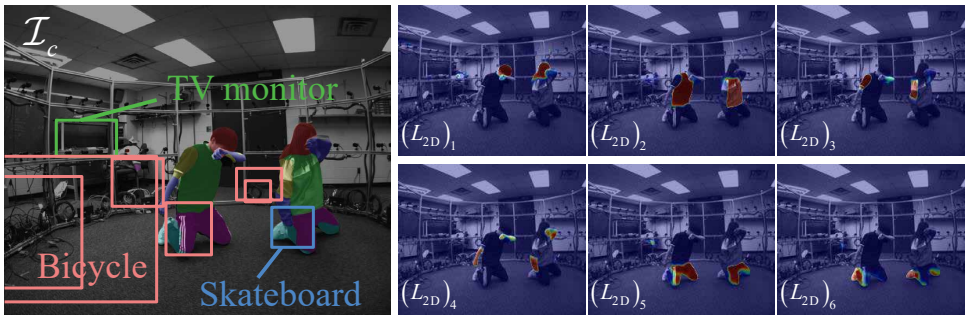


Figure 4.3. For each image $\mathcal{I}_c$, we use the recognition confidence (body segmentation [19]/object bounding box [20]) to build $L_{2D}(\mathbf{x}|\mathcal{I}_c)$ at each pixel $\mathbf{x}$ where the i^{th} element of L_{2D} is the likelihood (confidence) of the recognition for the i^{th} object class as shown on the right. For the illustration purpose, we only visualize the likelihood of body segments overlaid with the image while L_{2D} also includes object classes.

term affinity into a graph-cut formulation to infer the semantic labels jointly.

4.3.1 3D Semantic Map

We define the 3D semantic map, $L_{3D} \in [0,1]^N$, a probability distribution over semantic labels of a 3D trajectory. It is computed by reasoning about visibility and 2D recognition confidence at the 2D projections of the trajectory onto all cameras:

$$L_{3D}\left(\mathcal{X}\right) = \frac{1}{\Delta T}\sum_{t=T_e}^{T_d} \underset{c\in\mathcal{C}}{\text{Pool}}\left(L_{2D}\left(P\left(\mathbf{X}_t, c\right)|\mathcal{I}_c\right)\right), \tag{4.1}$$

where $\Delta T = T_d - T_e$ is the life span of the trajectory. The 3D trajectory label is evaluated at the 2D projection $P\left(\mathbf{X}_t, c\right)$ across all cameras over the trajectory life span. To alleviate noisy and coarse 2D recognition results, we introduce a view-pooling operation:

$$L_{c^*} = \underset{c\in\mathcal{C}}{\text{Pool}}\left(L_c\right) \quad \text{s.t.} \quad c^* = \underset{c\in\mathcal{C}}{\text{argmin}} \sum_{j=1}^{C} V_c \|L_c - L_j\|^2,$$

where we denote $L_{2D}\left(P\left(\mathbf{X}_t, c\right)|\mathcal{I}_c\right)$ as L_c, and $V(\mathbf{X}_t, c)$ as V_c by an abuse of notation. The view-pooling operation finds the best view among the visible cameras that is consistent with other view predictions (the weighted median of $\{L_c\}_{c\in\mathcal{C}}$).

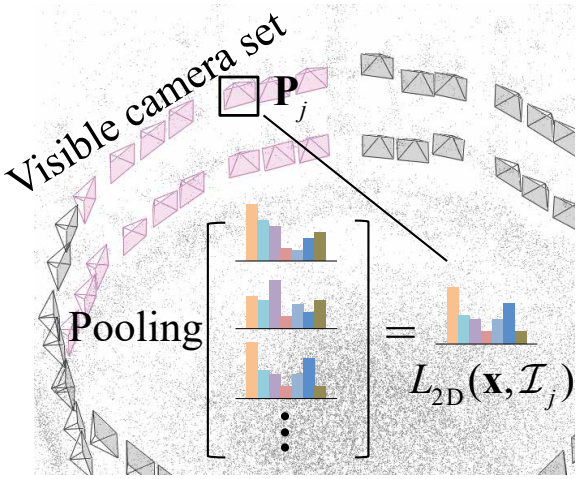


Figure 4.4. We construct the 3D semantic map $L_{3D}(\mathcal{X})$ via pooling L_{2D} over multiple views (view-pooling) by reasoning about visibility. The magenta camera is the visible camera set, and the bar graphs represent L_{2D}. The figures are best seen in color.

The view-pooling operation is based on our conjecture that among many views, there exist a few views that can confidently predict an object label. It is robust to noisy recognition outputs as shown in Figure 4.3 where many false positive bounding boxes are detected. The visibility based confidence measure can suppress inconsistent detection across views, and weighted median pooling can prevent from a view biased L_{3D}. This allows the pooled L_{2D} temporally consistent, which makes averaging over time meaningful.

Figure 4.4 illustrates the view-pooling operation over all cameras. A set of L_c (bar graphs) at the projected locations $\{P(\mathbf{X}, c)\}_{c \in \mathcal{C}}$ are used for the view-pooling that finds the L_{c^*} that best represents the distribution of L_c. For an illustrative purpose, we highlight the cameras that have high visibility with magenta color, *i.e.*, $V(\mathbf{X}, c) > \epsilon_e$.

4.3.2 3D Trajectory Affinity

An object that undergoes locally rigid motion provides a spatial cue to identify the affinity between fragmented trajectories. Consider two trajectories $\mathcal{X}_i$ and $\mathcal{X}_j$ that have overlapping lifetime, $\emptyset \neq \mathcal{S} = [T_e^i, T_d^i] \cap [T_e^j, T_d^j]$ where the superscript in T_e and T_d indicates the index of the trajectory. We measure the affinity of the trajectories as

follow:

$$A(i,j) = \exp\left(-\left(\|\mathbf{e}_i^j\|/\tau\right)^2\right) \tag{4.2}$$

where $A \in R^{M\times M}$ is an affinity matrix whose (i,j) entry measures the reconstruction error:

$$\mathbf{e}_i^j = \max_{t-1,t\in\mathcal{S}} \left\|\mathbf{X}_t^j - \mathbf{R}_t^i\mathbf{X}_{t-1}^j - \mathbf{t}_t^i\right\|.$$

$\mathbf{e}_i^j$ is the Euclidean distance between $\mathbf{X}_t^j$ and the predicted point by its emerging location $\mathbf{X}_{T_e}^j$ via its local transformation $(\mathbf{R}_t^i, \mathbf{t}_t^i) \in SE(3)$ (rotation and translation) learned by the i^{th} trajectory $\mathcal{X}_i$. This measure can be applied to long range trajectories, which establish a strong connection across an object, *e.g.*, left hand to left elbow trajectories. $i,j \in \mathcal{T} = \{1,\cdots,M\}$ where M is the number of trajectories. Unlike difference of pairwise point distance measure that has been used for trajectory clustering [122], our affinity takes into account general Euclidean transformation ($SE(3)$) that directly measures rigidity.

We learn the local transformation $(\mathbf{R}_t^i, \mathbf{t}_t^i)$ of the i^{th} trajectory at each time instant, given a set of neighbors:

$$\mathbf{R}_t^i = \Delta\mathbf{X}_t^{\mathcal{N}_i}\left(\Delta\mathbf{X}_{t-1}^{\mathcal{N}_i}\right)^{-1}, \quad \mathbf{t}_t^i = \mathbf{R}_t^i\mathbf{X}_{t-1}^i - \mathbf{X}_t^i \tag{4.3}$$

where $\Delta\mathbf{X}_t^{\mathcal{N}_i}$ is a matrix whose columns are made of relative displacement vectors of neighboring trajectories with respect to $\mathcal{X}_i$, *i.e.*, $\Delta\mathbf{X}_t^j = \mathbf{X}_t^j - \mathbf{X}_t^i$ where $j \in \mathcal{N}_i$ is the index of neighboring trajectories. The set of neighbors are chosen as

$$\mathcal{N}_i = \left\{ j \,\middle|\, \max_{t\in\mathcal{S}} \left\|\mathbf{X}_t^j - \mathbf{X}_t^i\right\| < \epsilon \right\},$$

where ϵ is the radius of a 3D Euclidean ball. Note that not all ϵ-neighbors belong to the same object which requires to evaluate the trajectory with Equation (4.2).

In practice, evaluating Equation (4.2) for all trajectories are computationally prohibitive. For example, it requires 10^{10} evaluations are needed for 100,000 trajectories[2] to fill in all entries in the affinity matrix A. Since it is unlikely that far distance trajectories belong to the same object class, we restrict the evaluations only for ϵ_a-neighbors ($\mathcal{N}_i^a$)

[2]In our experiments, the number of trajectories is order of $10^4 \sim 10^6$.

that are sufficient to cover a large portion of objects and greater ϵ, *e.g.*, $\epsilon_a = 30$cm and $\epsilon = 5$cm. Further, we randomly drop-out connections between neighboring trajectories for computational efficiency. This also increases the robustness of trajectory affinity that is often biased by the density of trajectories. When computing the local transformation in Equation (4.3), we embed RANSAC [234]: choosing random three trajectories from ϵ-neighbors and finding the local transformation that produces the maximum number of inliers.

4.3.3 Trajectory Label Inference

Inspired by multi-class pixel labeling using α-expansion [248], we infer the trajectory labels $U : \mathcal{T} \rightarrow \mathcal{L}$ where $\mathcal{L} = \{1, \cdots, N\}$ is the index set of object classes, by minimizing the following cost:

$$C(U) = \sum_{i \in \mathcal{T}} \phi(l_i, U(i)) + \lambda \sum_{i \in \mathcal{T}} \sum_{j \in \mathcal{N}_i^a} \psi(U(i), U(j)) \tag{4.4}$$

where λ is a hyper-parameter that control the weight between data ϕ and smoothness ψ costs.

The data cost can be written as:

$$\phi(l_i, U(i)) = \begin{cases} 0 & \text{if } \; l_i = U(i) \\ L_{3D}\left(\mathcal{X}_i\right)_{l_i} & \text{if } \; l_i \neq U(i) \end{cases},$$

where it penalizes the discrepancy between the 3D semantic map predicted by a series of 2D recognitions and assigned label. $L_{3D}\left(\mathcal{X}_i\right)_{l_i}$ is the l_i^{th} entry of L_{3D} that measures the likelihood of $\mathcal{X}_i$ being class l_i.

The smoothness cost can be described by the trajectory affinity:

$$\psi(U(i), U(j)) = \begin{cases} 0 & \text{if } \; U(i) = U(j) \\ A(i,j) & \text{if } \; U(i) \neq U(j) \end{cases},$$

where it penalizes the label difference between trajectories that undergo the same local rigid transformation. l_i is the label index computed from L_{3D}:

$$l_i = \underset{l \in \mathcal{L}}{\text{argmax}} \;\; L_{3D}\left(\mathcal{X}_i | \{\mathbf{P}_c, \mathcal{I}_c\}_{c \in \mathcal{C}}\right).$$

Due to multi-class labeling, minimization of Equation (4.4) is highly nonlinear while the iterative α-expansion algorithm has been shown a strong convergence towards the global minimum [248, 249].

4.4 3D Trajectory Reconstruction

In this section, we describe the procedure of the 3D trajectory reconstruction algorithm modified from Joo et al. [121] to produce denser and more accurate trajectories. **(1) Camera calibration** We calibrate the intrinsic parameter of each camera (focal length, principal points, and radial lens distortion), independently, and use standard structure from motion to calibrate extrinsic parameters (relative rotation and translation). In the bundle adjustment, the extrinsic and intrinsic parameters are jointly refined. To accelerate further image based matching, we learn the image connectivity graph [247] $\mathcal{G}_m = (\mathcal{V}_m, \mathcal{E}_m)$ through exhaustive pairwise image matching, *e.g.*, two cameras that have more than 90 degree apart are unlikely to match to each other. **(2) Point cloud triangulation** At each time instant, we find dense feature correspondences using grid-based motion statistics (GMS) [250] among $\mathcal{G}_m$ and triangulate each 3D point $\mathbf{X}$ with RANSAC. The initial visibility for the c^{th} camera is set to $V(\mathbf{X}, c) = \exp(-(\|P(\mathbf{X}, c) - x(c)\|/\sigma)^2)$ where the σ is the tolerance of the reprojection error and $x(c)$ is the corerspondecne point at camera c. **(3) 3D point tracking** The triangulated points are used for build trajectory stream. For each point $\mathbf{X}_t$ at the t time instant, we project the point onto the visible set of cameras, *i.e.*, $P(\mathbf{X}_t, c \in \mathcal{V})$ where $\mathcal{V} = \{j | V(\mathbf{X}_{t-1}, c) > \epsilon_s\}$ where ϵ_s is the threshold for the probability of visibility. These projected points are tracked in 2D using optical flow and triangulated with RANSAC to form $\mathbf{X}_{t+1}$. Similar to the visibility initialization, the probability of visibility $V(\mathbf{X}_{t+1}, c)$ is updated using reprojection error. We iterate this process (tracking→triangulation→visibility update) until the average reprojection is higher than 2 pixels or the number of visible cameras $|\mathcal{V}|$ is less than 2.

4.5 Validation

To validate our semantic trajectory reconstruction algorithm, we evaluate on real-world datasets collected by multi-camera system.

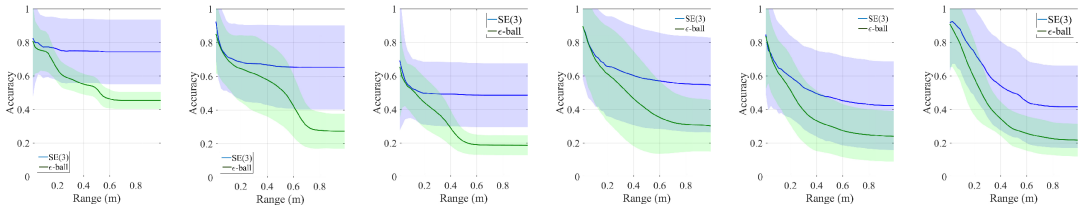
4.5.1 Human Interaction Dataset

9 new vignettes that include diverse human interactions are captured: **Pet interaction**: A dog owner naturally interacts with her dog: ask him to sit, turn around and jump. The dog also plays with his doll and seek snack while walking around with the owner. This pet interaction demonstrates strength of our system, *i.e.*, reconstructing fine detailed interactions, not limited to humans [122]; **International Latin ballroom dance**: Two sport dancers practice for Cha-cha style dance competition where the physical interactions between them are highly stylized. The dancers wear textureless black suit and skirt where semantic labeling is likely noisy; **K-Pop group dance**: Two experienced K-Pop dancers perform the group break dance. The dances are designed to be synchronized, jerky, and fast; **Object manipulation**: Two students manipulate various objects such as doll, flowerpot, monitor, umbrella, and hair drier in a cluttered environments. This vignette demonstrates that the system is able to handle multiple objects; **Bicycle riding**: A person rides a bicycle that induces large displacement. This interaction introduces significant occlusion, *i.e.*, the person is a part of the bicycle; **Tennis swing**: A person practices fore- and back-hand strokes with a tennis racket. The tennis racket is often difficult to detect as the racket head is mostly transparent; **Basketball I**: A student player practices dribbling which includes fast ball motion; **Basketball II**: An other player tries to block the opponent's motion that includes severe occlusion between players.

4.5.2 Quantitative Evaluation

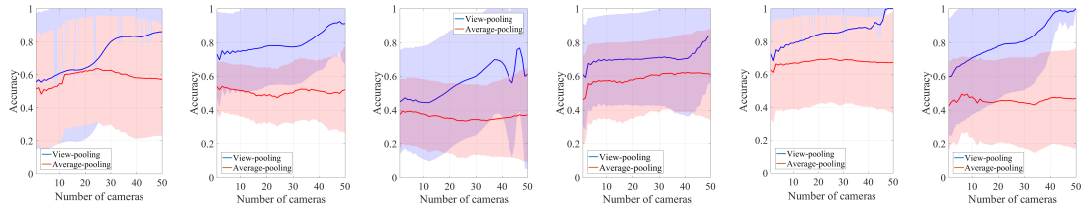
We quantitatively evaluate our representation and algorithm in terms of three criteria: (1) robustness of 3D semantic map (view-pooling); (2) effectiveness of the affinity measure; and (3) predictive validity of semantic labels where all datasets are used for the evaluations. Note that as no ground truth data or benchmark dataset is available, we conduct ablation studies to validate our methods.

Robustness of 3D semantic map We introduce the view-pooling operation that takes the weighted median of recognition confidence based on visibility. This operation allows robustly predicting the 3D semantic map L_{3D} as it is not sensitive to erroneous



(a) Pet int. (b) L. dance (c) K-Pop (d) Bike (e) Tennis (f) Basketball II

Figure 4.5. We evaluate the effectiveness of our affinity map computed by estimating local Euclidean transformation SE(3). While the effectiveness of ϵ_s-neighbors diminishes rapidly after 10 cm, our method still holds for longer range, *e.g.*, 1 m.



(a) Pet int. (b) L. dance (c) K-Pop (d) Tennis (e) Basketball I (f) Basketball II

Figure 4.6. We evaluate semantic label prediction via an ablation study: to use a subset of cameras to assign the semantic labels to the trajectories and validate the labels by comparing the labels of projections with the held-out images. Our view-pooling method outperforms the average-pooling with large margin for all sequences.

detection. To evaluate its robustness, we measure the temporal consistency of the view-pooling operation along a trajectory. Ideally, the view-pooled recognition confidence should remain constant across time as it belongs to the trajectory of the same object. We compare the view-pooling with average-pooling across randomly all cameras using normalized correlation measure across time, *i.e.*, $NC(L^0_{\text{vp}}, L^t_{\text{vp}})$ where L^t_{vp} is the view-pooled recognition confidence at the t time instant. We summarize the results on all sequences in Table 4.1. Our method shows a graceful degradation as time progress up to 15 seconds while the average-pooling is highly biased by noisy recognition, which produces drastic performance gradation (no temporal coherence).

Time (second)	1s	3s	5s	7s
View pool	**0.96**±0.01	**0.90**±0.02	**0.89**±0.03	**0.88**±0.02
Ave. pool	0.43±0.10	0.44±0.10	0.43±0.10	0.48±0.09
Time (second)	9s	11s	13s	15s
View pool	**0.89**±0.02	**0.88**±0.03	**0.87**±0.05	**0.79**±0.08
Ave. pool	0.44±0.09	0.43±0.10	0.42±0.10	0.37±0.10

Table 4.1. Time consistency of 3D semantic map

Effectiveness of affinity measure We compute the affinity based on local transformation per trajectory. This method is highly effective to relate with long term fragmented trajectories. We compare the validity of our affinity measure with that of ϵ_s-neighbors ($\mathcal{N}_s$), *i.e.*, the distance between trajectories over time remains less than ϵ_s. To evaluate, two neighboring trajectories for both methods are randomly chosen and projected onto cameras. Concretely, we measure $\sum_{j \in \mathcal{N}_s} E(i,j)$ where

$$E(i,j) = \begin{cases} 0 & \text{if } L(P(\mathbf{X}_t^i, c)|\mathcal{I}_c) = L(P(\mathbf{X}_t^j, c)|\mathcal{I}_c) \\ 1 & \text{otherwise} \end{cases} .$$

$L : R^2 \rightarrow \mathcal{L}$ outputs the semantic label index given the 2D projection. If the measure is small, it indicates that the neighbors are correctly identified. Figure 4.5 illustrates the comparison over 6 different sequences. Each one has different global and local motion. If the motion is largely global, the affinity measure can confuse as multibody motion is identified as a rigid body motion as shown in Basketball II. Nonetheless, our method outperforms the ϵ_s-neighbors for all sequences. In particular, it shows much stronger performance at long range trajectories (0.6-1 m), which makes the large scale label inference possible.

Predictive validity of 3D semantic label We evaluate the semantic label inference via cross validation scheme. We label a 3D trajectory with a subset of cameras and project onto the held-out camera to evaluate the predictive validity. Ideally, the trajectory label should be consistent with any view as visibility is considered, and therefore, the projected label must agree with the recognition result. As we infer the semantic labels of the trajectories jointly by consolidating multiple view recognition, the number of cameras plays a key role in the inference. We test the predictive validity by changing the number of cameras to label trajectories as shown in Figure 4.6. When the number

	R.motion	B.ball I	Latin	K-Pop	Pet	Bike	Tennis
Joo et al. [122]	0.8547	0.8862	0.7532	0.5019	0.4819	0.5307	0.7317
AP(1)	0.7532	0.6271	0.5388	0.3730	0.5145	0.5297	0.4607
AP(30)	0.8578	0.6879	0.5014	0.3431	0.6276	0.6341	0.6029
AP(69)	0.8584	0.7309	0.7769	0.5706	0.6018	0.6162	0.6691
VP(1)	0.8403	0.7259	0.7307	0.4485	0.5755	0.7432	0.6099
VP(30)	0.9092	0.8650	0.7753	0.5992	0.8015	0.7064	0.7133
VP(69) [**Ours**]	**0.9326**	**0.9572**	**0.8753**	**0.6985**	**0.8132**	**0.8394**	**0.8438**

Table 4.2. We compare our method with multiple baselines in terms of accuracy. AP(x) and VP(x) refer to average-pooling and view-pooling, respectively where x is the maximum number of visible cameras.

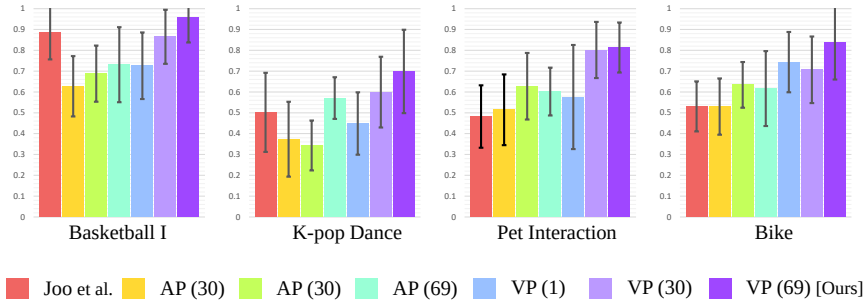
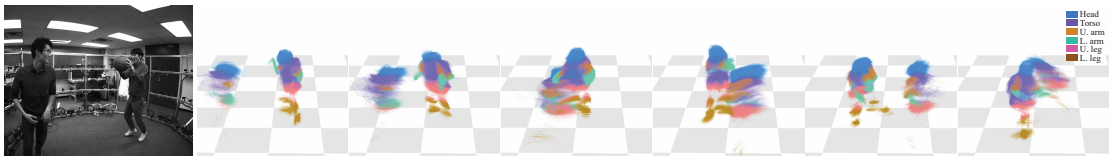
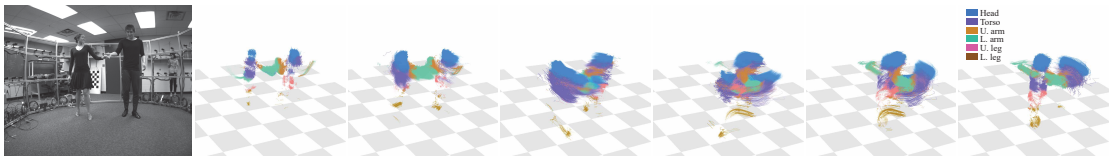


Figure 4.7. Our method outperforms all baselines. The notation, AP(x) and VP(x) are consistent with in Table 4.2

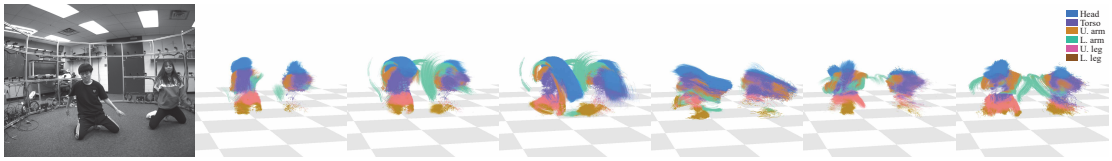
of cameras is few, *e.g.*, 1-5, our method using view-pooling performs similarly with average-pooling. However, the performance quickly is boosted as the number of camera increases, *i.e.*, in most cases, it produces more than 0.6 accuracy at 20 cameras for inference. In Table 4.2, we further compare our method with the approach from Joo et al.[122], where the semantic label on the trajectory is inferred by 3D human body anatomical key-points. As highlighted in Figure 4.7, our method outperforms [122] in all possible scenarios (*e.g.*occlusion, dynamic deformation, object interaction, multiple people).



(a) Basketball II



(b) Latin dance



(c) K-Pop

(d) Tennis

Figure 4.8. Qualitative evaluation. Best seen in color. For an illustrative purpose, the last 30 frames of the trajectories are visualized.

4.5.3 Qualitative Evaluation

We apply our method to reconstruct dense semantic trajectories in 3D as shown in Figure **??** 4.9, and 4.8. The colors of the trajectories indicate the semantic labels.

4.6 Summary

We present an algorithm to reconstruct semantic trajectories in 3D using a large scale multi-camera system. This problem is challenging because of fragmented trajectories and noisy/coarse recognition in 2D. We introduce a new representation to encode the

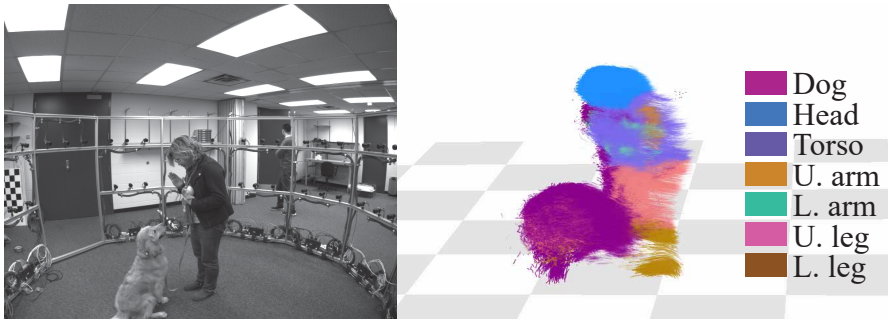


Figure 4.9. Pet interaction

visual semantics to each trajectory called 3D semantic map that allows us to consolidate multiple view noisy recognition results by leveraging view pooling based on their visibility and recognition confidence. 3D spatial relationship between fragmented trajectories is modeled by local rigid transformation that can establish the connection between long range trajectories. These two cues are integrated into a graph-cut formulation to infer precise labeling of the trajectories. Note that Our framework is not specific to the choice of the 2D recognition models.

Chapter 5

Learning to Reconstruct 3D Face Model from a Single Image

High-fidelity face models enable the building of realistic avatars, which play a key role in communicating ideas, thoughts and emotions. Thanks to the uprising of data-driven approaches, highly realistic and detailed face models can be created with active appearance models (AAMs) [126, 127], 3D morphable models (3DMMs) [128], or deep appearance models (DAMs) [60]. These data-driven approaches jointly model facial geometry and appearance, thus empowering the model to learn the correlation between the two and synthesize high-quality facial images. Particularly, the recently proposed DAMs can model and generate realistic animation and view-dependent textures with pore-level details by leveraging the high capacity of deep neural networks. However, driving these realistic face models requires special input data, *e.g.*, 3D meshes and unwrapped textures. In this chapter, we present a method to high-fidelity face tracking using a monocular camera by learning 3D face model prior.

5.1 3D Face Model Reconstruction from a Single Image

Many face models including DAMs can be viewed as an encoder and decoder framework. The encoder E_X takes an input $X = (\mathbf{G}, \mathbf{T})$, which corresponds to the geometry and unwrapped texture, respectively. $\mathbf{G} \in \mathbb{R}^{G \times 3}$ represents the 3D locations of G vertices which form a 3D mesh of the face. Note that rigid head motion has already been

removed from the vertex locations, *i.e.*, $\mathbf{G}$ represents only local deformations of the face. The unwrapped texture $\mathbf{T} \in \mathbb{R}^{T \times T \times 3}$ is a 2D image that represents the appearance at different locations on $\mathbf{G}$ in the UV space. The output of E_X is the intermediate code $\mathbf{z}$. The decoder D then takes $\mathbf{z}$ and computes a reconstructed output $\widetilde{X} = \mathrm{D}(\mathbf{z}) = \mathrm{D}(\mathrm{E}_X(X))$. The encoder and decoder are learned by minimizing the difference between X and $\widetilde{X}$ for a large number of training samples.

The challenge is that $X = (\mathbf{G}, \mathbf{T})$, *i.e.*, the 3D geometry and unwrapped texture, is not readily available in a monocular image $\mathbf{I}$. Therefore, we learn a separate deep encoder called *I2ZNet* (Image-to-$\mathbf{z}$ network): $(\mathbf{z}, \mathbf{H}) \leftarrow \mathrm{E}_{\mathbf{I}}(\mathbf{I})$, which takes a monocular image $\mathbf{I}$ as input and directly outputs $\mathbf{z}$ and the rigid head pose $\mathbf{H}$. I2ZNet first extracts the domain independent two-stream features using the pre-trained VGGNet [16] and HourglassNet [251], which provides perceptual information and facial landmarks, respectively. The multiple depth-level two-stream features are combined with skip connections, and are regressed respectively to the intermediate representation $\mathbf{z} \in \mathbb{R}^{128}$ and the head pose $\mathbf{H} \in \mathbb{R}^6$ using several fully connected layers [252]. This architecture allows to directly predicts the parameters ($\mathbf{z}$, $\mathbf{H}$) based on the category-level semantic information from the deep layers and local geometric/appearance details from the shallow layers at the same time. $\mathbf{z}$ can be given to the existing decoder D to decode the 3D mesh and texture, while $\mathbf{H}$ allows to reproject the decoded 3D mesh onto the 2D image. Figure 5.1 illustrates the overall architecture of I2ZNet, and more details are described in the Section 5.2.

$\mathrm{E}_{\mathbf{I}}$ is trained in a supervised way with multiview image sequences used for training E_X and D of DAMs. The by-product of learning E_X and D are the latent code $\mathbf{z}_{gt}$ and the head pose $\mathbf{H}_{gt}$ at each time. As a result of DAM training, we acquire as many tuples of $\{\mathbf{I}_v, \mathbf{z}_{gt}, \mathbf{H}_{gt}\}$ as the camera views $\{v\}$ at every time t as training data for $\mathrm{E}_{\mathbf{I}}$.

The total loss to train $\mathrm{E}_{\mathbf{I}}$ is defined as

$$L_{\mathrm{E}_{\mathbf{I}}} = \lambda_{\mathrm{z}} L_{\mathrm{z}} + \lambda_{\mathrm{H}} L_{\mathrm{H}} + \lambda_{\mathrm{view}} L_{\mathrm{view}}, \tag{5.1}$$

where L_{z} and L_{H} are the losses for $\mathbf{z}$ and $\mathbf{H}$, respectively, and L_{view} is the view-consistency loss. λ_{z}, λ_{H} and λ_{view} are weights for L_{z}, L_{H} and L_{view}, respectively.

L_{z} is the direct supervision term for $\mathbf{z}$ defined as

$$L_{\text{z}} = \sum_{v,t} \left\| \mathbf{z}_{\mathbf{I}_v^t} - \mathbf{z}_{gt}^t \right\|_2^2, \tag{5.2}$$

where $\mathbf{z}_{\mathbf{I}}$ is a DAM latent code regressed from $\mathbf{I}$ via $\text{E}_{\mathbf{I}}$.

Inspired by [139, 253], we formulate $L_{\mathbf{H}}$ as the reprojection error of the 3D landmarks predicted via $\text{E}_{\mathbf{I}}$ w.r.t. the 2D ground-truth landmarks $\mathbf{K}_{gt} \in \mathbb{R}^{K\times 2}$ for the head pose prediction:

$$L_{\text{H}} = \frac{1}{K} \sum_{k,v,t} \left\| \mathbf{\Pi} \mathbf{H}_{\mathbf{I}_v^t} \mathbf{K}^k(\mathbf{G}_{\mathbf{I}_v^t}) - \mathbf{K}_{gt}^k \right\|_2^2, \tag{5.3}$$

where K is the number of landmarks, $\mathbf{\Pi} = [1\ 0\ 0; 0\ 1\ 0]$ is a weak perspective projection matrix, and $\mathbf{H}_{\mathbf{I}}$ is the head pose regressed from $\mathbf{I}$ via I2ZNet. $\mathbf{G}_{\mathbf{I}}$ is the set of vertex locations decoded from $\mathbf{z}_{\mathbf{I}}$ via D, and $\mathbf{K}^k(\cdot)$ computes the 3D location of k-th landmark from $\mathbf{G}_{\mathbf{I}}$.

Because the training image data is captured with synchronized cameras, we want to ensure that the regressed $\mathbf{z}$ is the same for images from different views captured at the same time. Therefore, we incorporate the view-consistency loss L_{view}, defined as

$$L_{\text{view}} = \sum_{v,w,t} \left\| \mathbf{z}_{\mathbf{I}_v^t} - \mathbf{z}_{\mathbf{I}_w^t} \right\|_2^2. \tag{5.4}$$

We randomly select two views at every training iteration.

5.2 I2ZNet

In this section, we detail the architecture of I2ZNet.

5.2.1 Inputs and Outputs

Given a cropped input face image $\mathbf{I} \in \mathbb{R}^{256\times 256\times 3}$, the I2ZNet directly predicts the low-dimensional facial state codes $\mathbf{z} \in \mathbb{R}^{128}$, and a set of head pose parameters $\mathbf{H} \in \mathbb{R}^6 = \{f,\ r_x,\ r_y,\ r_z,\ t_x,\ t_y\}$, where $\mathbf{f} = \{f\},\ \mathbf{r} = \{r_x,\ r_y,\ r_z\},\ \mathbf{t} = \{t_x,\ t_y\}$ are focal length scale, Euler angle, and 2D translation respectively. The pre-trained decoder D decodes $[\mathbf{z}^{\mathsf{T}}, \mathbf{H}^{\mathsf{T}}]$ to generate high fidelity 3D face geometry $\mathbf{G} \in \mathbb{R}^{7306\times 3}$ and view dependent texture map $\mathbf{T} \in \mathbb{R}^{1024\times 1024\times 3}$. Note that, we are using the same decoder with [60], while we replace its encoder network E_X with our I2ZNet.

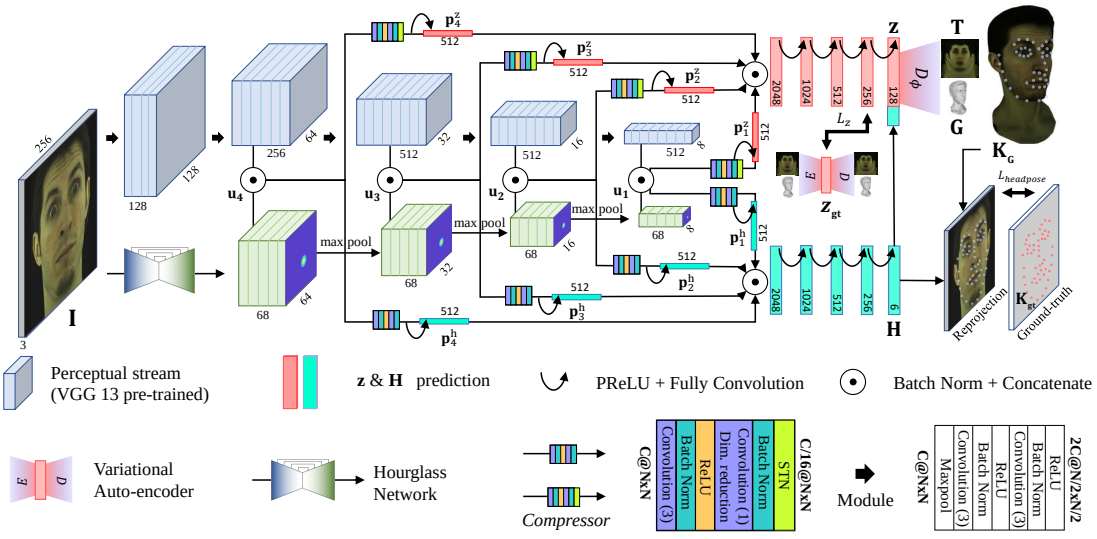


Figure 5.1. *Z-adaptor* directly regresses the latent facial state codes **z** and headpose **H** from a face image **I**, and the pre-trained decoder D generates full 3D face geometry and high resolution texture.

5.2.2 Domain Invariant Multi-level Unified Features

Given an input image **I**, I2ZNet extracts the features from two-stream networks: VGGNet [16] and HourglassNet [251]. VGGNet captures perceptual information such as facial details or shape, while HourglassNet guides "where to look" by providing facial geometry features, e.g. facial landmark heatmaps. We complete the multi-level unified features $\mathbf{u}_l \in \mathbb{R}^{(32*2^l)\times(32*2^l)\times ch_l}$ by concatenating the two-stream features, where $l = \{4, 3, 2, 1\}$ denotes the feature depth-level and the associated channel size is $ch_l \in CH = \{324, 580, 580, 580\}$. Here, we simply max-pool the output from HourglassNet to make the feature size equal to each level of VGG feature. The feature scale inconsistency between two different networks (VGGNet and HourglassNet) is resolved by normalization layer before concatenation. Our multi-level unified features are more domain (color, illumination, or head pose) invariant by learning from domain generalized datasets [254, 255]. Note that, the pre-learned weights on the two-stream networks are fixed in the following training steps such that we prevent I2ZNet from being domain specific.

5.2.3 Latent Parameter Regression

Inspired by many recent papers [256, 252] which have proposed the use of combination of deep and shallow features to capture semantic-level information and local appearance details at the same time, we concatenate feature vectors from each depth level $\mathbf{p}^z_{4..1}$, $\mathbf{p}^h_{4..1} \in \mathbb{R}^{512}$, which are encoded from $\mathbf{u}_{4..1}$, and they are respectively regressed to $\mathbf{z}$ and $\mathbf{H}$ using several fully connected layers. Here, however, it requires very heavy computational costs for converting three-dimensional features $\mathbf{u}_l$ to single dimensional one $\mathbf{p}^{z,h}_l$ in a fully connected way. Similar to [252], we alleviate this bottleneck by channel-wise feature compression of $\mathbf{u}_l$ to one-sixteenth of its original channel size using two convolutional layers as described as *Compressor* layer in Figure 5.1.

5.3 Validation

We introduced the domain and view invariant property of our network. To gain more insight to our model, we perform following ablation experiments.

5.3.1 Ablation Study on I2ZNet Structure

To validate the performance gain of each component on our regression network, we compare I2ZNetagainst three baseline networks: **VGG+Skip+Key** denotes I2ZNet, which uses VGGNet, multi-level features (skip connections), and landmarks from HourglassNet. **VGG+Skip**: landmarks guidance is removed. **VGG**: Multi-level features (skip connection) are further removed and only deep features are used for regression. **VGG Scratch** has the same structure with **VGG** but it is trained from scratch. For other settings which use **VGG**, pre-trained VGG-16 features are used, and the VGG portion of the network is not updated during training. The models are tested on unseen test datasets where the vertex-wise dense ground-truth is available. Three metrics are employed to evaluate performance: (1) accuracy for geometry is computed by Euclidean distance between predicted and ground-truth 3D vertices, (2) accuracy for texture is calculated by pixel intensity difference between predicted and ground-truth texture, and (3) the temporal stability is measured by:

	VGG Scratch	VGG	VGG+Skip	VGG+Skip+Key
Geometry	1.011	1.481	0.411	**0.315**
Texture	0.016	0.027	0.007	**0.004**
Temporal	2.143	3.138	1.499	**1.446**

Table 5.1. Ablation test on I2ZNet. The average score with respect to all subjects are reported.

$$\frac{1}{G}\sum_{i=1}^{G}\frac{\left\|\mathbf{G}_i^{t+1}-\mathbf{G}_i^{t}\right\|_2+\left\|\mathbf{G}_i^{t}-\mathbf{G}_i^{t-1}\right\|_2}{\left\|\mathbf{G}_i^{t+1}-\mathbf{G}_i^{t-1}\right\|_2}, \tag{5.5}$$

where $\mathbf{G}_i^t$ corresponds to the 3D location of vertex i at time t. This metric assumes that the vertices of the 3D mesh should move on a straight line over the course of three frames, thus unstable or jittering predictions will lead to higher (worse) score. The lowest (best) metric score is 1.

The average scores with respect to the four test subjects are reported in Table 5.1, and the representative subject results are visualized in Figure 5.2. We observe that multi-level features (**VGG+Skip**) significantly improves performance over **VGG**, while adding keypoints (**VGG+Skip+Key**) further improves performance. **VGG** seems to lack of capacity to directly regress the latent parameters with only pre-trained deep features which are not updated. More ablation studies (*e.g.*, tests on view consistency and robustness to the synthetic visual perturbation) on I2ZNet are described in the supplementary manuscript.

5.3.2 Robustness to Visual Perturbation

We introduced the domain and view invariant property of our network. To verify this, we test I2ZNet on four different scenarios, **View**, **Color**, **Light**, and **Jitter**, where the baseline networks are the same with the ones described in Section 5.3.1.

View represents the test dataset of multiview videos, where they are accurately synchronized and thus I2ZNet should predict the same facial local deformation to make

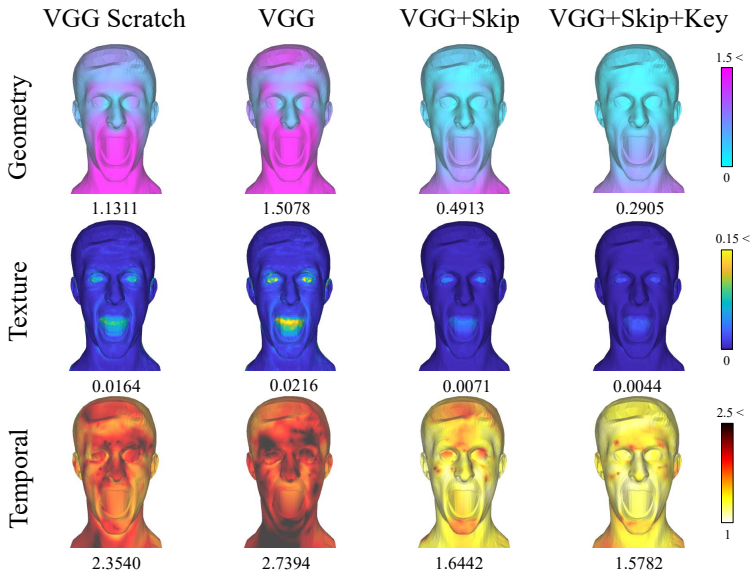


Figure 5.2. Ablation test on I2ZNet with a representative subject. The vertex-wise error is visualized with the associated average score for subject 1.

the facial configuration consistent across the views. To verify this view consistent prediction ability, we pick the most central camera as a ground-truth view and evaluate the performance of other views. We use simple vertex-wise Euclidean distance between the 3D faces predicted from central view and other views meaning that the lower score shows better consistency. The overall performance is summarized in Table 5.2 and Figure 5.3, where the proposed network outperforms all other baselines. We can further notice that the combination of skip connection and landmark guidance helps the network to figure out the facial geometry configuration when predicting the facial configuration from different views based on the comparison of **VGG** with **VGG+Skip** and **VGG+Skip+Key**. Note that, when evaluating the view consistency, we remove the texture and head pose from a predicted 3D face because they have view dependent property in our system.

Color, **Light**, **Jitter**, and **Background** represent video sequences which contain synthetic perturbation with random color, gamma, jitters by similarity transformation (scale, rotation, and translation variation), and white dotted background noise. The goal of the test on these sequences is to verify the domain generality. For example, if

		View	Color	Light	Jitter	Background
VGG Scratch	Geometry	0.607	1.485	1.175	0.983	1.285
	Headpose	-	17.48	6.965	-	15.84
	Texture	-	0.021	0.014	0.016	0.015
VGG	Geometry	1.352	1.258	1.510	1.736	1.076
	Headpose	-	16.61	13.98	-	16.42
	Texture	-	0.020	0.021	0.025	0.016
VGG +Skip	Geometry	0.3967	0.622	0.227	1.331	0.669
	Headpose	-	2.579	0.728	-	8.750
	Texture	-	0.009	0.003	0.018	0.009
VGG +Skip +Key	Geometry	**0.255**	**0.505**	**0.151**	**0.896**	**0.417**
	Headpose	-	**1.676**	**0.684**	-	**8.172**
	Texture	-	**0.007**	**0.002**	**0.012**	**0.006**

Table 5.2. Ablation studies on I2ZNet.

I2ZNet outputs a completely different 3D facial configuration given a perturbed image comparing to the one before the perturbation, then it implies that the network is overfitted to the training data domain. Therefore, we evaluate the performance of I2ZNet on the sequence after the perturbation in light of the results from the ones before the perturbation. To measure this relative accuracy, we employ three metrics: geometry, texture, and head pose. For geometry and texture, we simply calculate the 3D distance and color difference of the 3D faces. For head pose, we measure the 2D distance between the ground-truth points and the reprojection of the vertices on the 3D face to the input with the predicted head pose. The average scores with respect to the entire test subjects (4 subjects) are reported in Table 5.2, and the representative subject results are visualized in Figure 5.3. From the comparison of **VGG Scratch** with **VGG+Skip+Key**, we can notice that the pre-trained nature of the feature extraction parts (VGGNet and HourglassNet) plays a key role to avoid overfitting from a specific domain. Further, the comparison between **VGG+Skip** and **VGG+Skip+Key** implies that the landmark module guides the attention of the network such that it prevents from the network distraction even under the background perturbation.

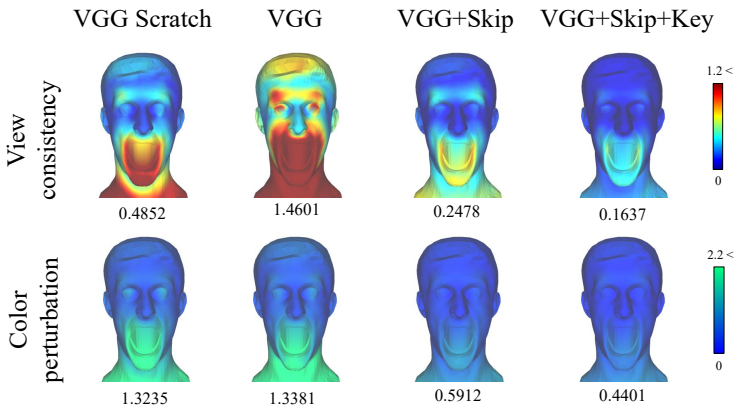


Figure 5.3. Visualization of the vertex-wise accuracy with a representative subject for ablation studies on view consistency and color sensitivity. The average score is reported for each metric, where the lower score shows the better performance for both scenarios.

5.4 Summary

We introduced a method to build human face prior to infer a complete face model from a single image. We proposed a novel deep neural network that predicts the intermediate representation and head pose of a high-fidelity 3D face model from a monocular image. The reconstructed 3D face model that describes the underlying geometry and the associated view-dependent appearance allows us to render the person from different viewpoints. We demonstrated that our 3D face prior is robust to various visual perturbation such as color, viewpoint changes, and background clutter due to the domain-invariant nature of the deep feature representation, which are validated by the ablation study.

Part II

Learning to Adapt the Learned 3D Avatars to General Unconstrained Scenes

Chapter 6

Self-Supervised Adaptation of High-Fidelity 3D Face Model

In Chapter 5, we modeled data-driven prior of high-fidelity 3D face model using large amount of multiview data captured from controlled lab environment. This allows us to predict a complete 3D face model, from a single image. Unfortunately, barriers exist when applying this 3D face prior to in-the-wild imagery due to *domain mismatch*. Domain mismatch refers to the fact that the visual statistics of in-the-wild imagery are considerably different from that of a controlled lab environment used to build the high-fidelity face model. In-the-wild imagery includes various background clutter, low resolution, and complex ambient lighting as shown in Figure 6.1. Such domain gap breaks the correlation between appearance and geometry learned by the data-driven model and the prior may no longer work well in the new domain. This challenge greatly inhibits the wide-spread use of the high-fidelity face models using a single camera. In this chapter, we overcome this challenging, we present a self-supervised domain adaptation technique that can adapt the learned prior to new environments without requiring any labeled data from the new domain.

6.1 Handling Domain Mismatch

In Chapter 5, we formulate the data-driven prior of high-fidelity face model, i.e., Deep Appearance Models (DAMs [60]), by designing and training a new regression model,

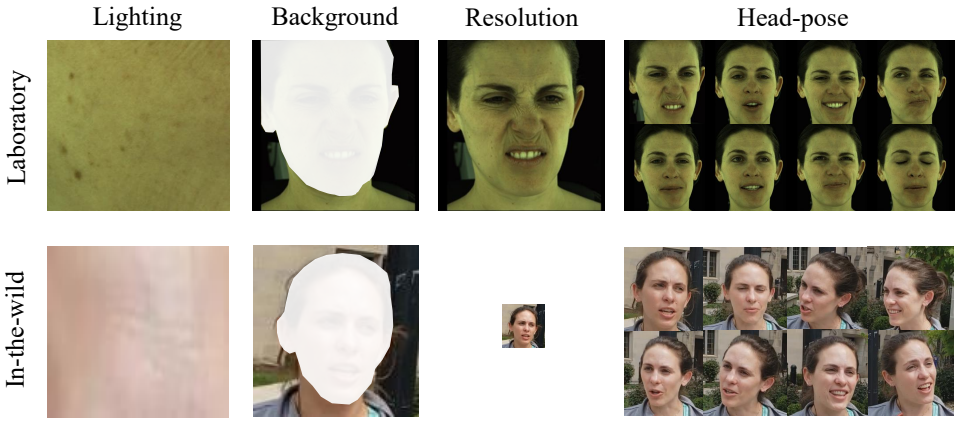


Figure 6.1. The comparison of the images captured from controlled laboratory environment and in-the-wild environment. Domain gap exists (lab vs. in-the-wild) in terms of lighting (consistent vs. ambient), background (black vs. cluttered), and image resolution (high vs. low), and a subject's headpose (static vs. dynamic).

called I2ZNet, that can predicts the intermediate face representation of DAMs from a single image. In this chapter, we adapt I2ZNet $\mathrm{E}_{\mathbf{I}}$ to a new in-the-wild domain using a set of unlabeled images directly from the new domain in a self-supervised manner. A high-level overview is shown in Figure 6.2.

The domain adaptation refines the encoder $\mathrm{E}_{\mathbf{I}}$ by minimizing Eq. (6.1), which consists of (1) consecutive frame texture consistency L_{CFTC}, (2) model-to-observation texture consistency L_{MOTC}, and (3) facial landmark reprojection consistency L_{FLRC}:

$$\sum_{t} \lambda_z L_z^t + \lambda_{\mathrm{CFTC}} L_{\mathrm{CFTC}}^t + \lambda_{\mathrm{MOTC}} L_{\mathrm{MOTC}}^t + \lambda_{\mathrm{FLRC}} L_{\mathrm{FLRC}}^t, \tag{6.1}$$

where λ_z, λ_{CFTC}, λ_{MOTC} and λ_{FLRC} correspond to the weights for each loss term. The consecutive frame texture consistency loss is our key contribution. It adapts $\mathrm{E}_{\mathbf{I}}$ such that textures computed from predicted geometry are temporally coherent. The model-to-observation texture consistency avoid drift of prediction errors over time via pixel-wise matching between DAM generated texture and observed texture.

Consecutive Frame Texture Consistency

Inspired by the brightness constancy assumption employed in many optical flow algorithms, we can reasonably assume that 3D face tracking for two consecutive frames is

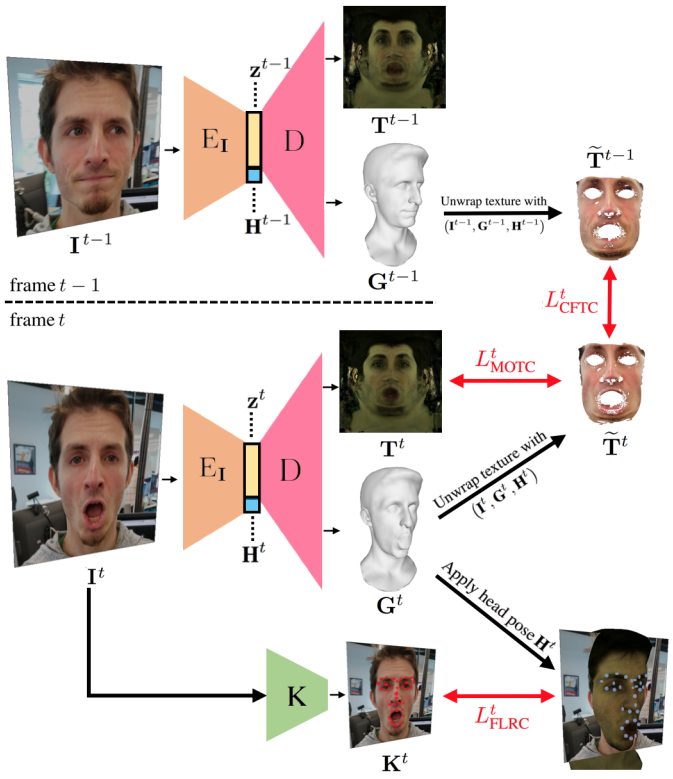


Figure 6.2. Overview of our proposed self-supervised domain adaptation process. For two consecutive frames, we run $\mathrm{E_I}$ followed by D to acquire the geometry and texture. The head-pose detector is also run to compute head-pose. Then, the geometry, input image, and head pose is used to compute the unwrapped texture $\widetilde{T}$. This enables us to compute L_{CFTC} and L_{MOTC}. For frame t, we run facial landmark detection, which is then used to compute L_{FLRC}. These losses can then back-propagate gradients back to $\mathrm{E_I}$ to perform self-supervised domain adaptation.

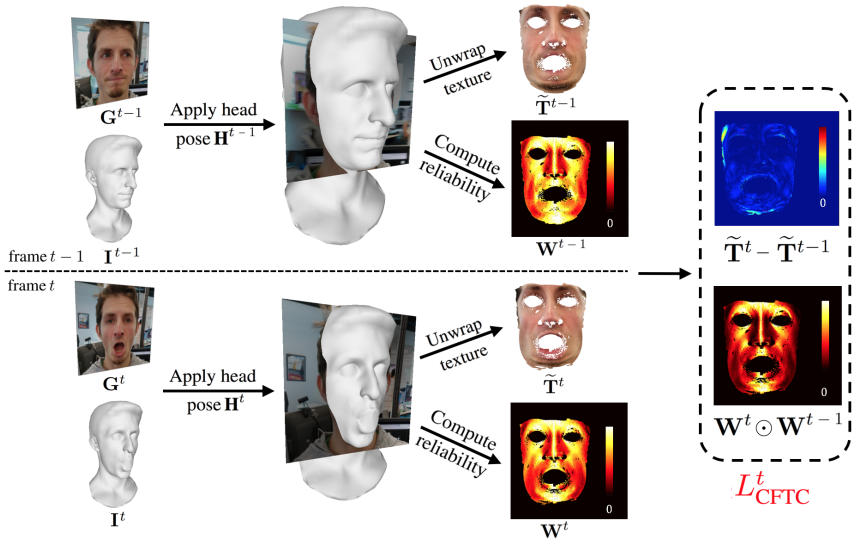


Figure 6.3. Steps and intermediate results for computing CFTC loss.

accurate only if unwrapped textures for the two frames are nearly identical. Inversely, if we see large changes in unwrapped texture across consecutive frames, then it is highly likely due to inaccurate 3D geometry predictions. We make the assumption that environmental lighting and the appearance of the face does not change significantly between consecutive frames, which is satisfied in most scenarios. Otherwise, we do not make any assumptions on the lighting environment of a new scene, which makes our method more generalizable than existing methods which, for example, approximates lighting with spherical harmonics [139].

The consecutive frame texture consistency loss L_{CFTC} is defined as

$$L^{t}_{\text{CFTC}} = \frac{1}{W^{t,t-1}} \sum_{i,j} (\mathbf{W}^{t} \odot \mathbf{W}^{t-1})_{ij} \left\| \widetilde{\mathbf{T}}^{t}_{ij} - \widetilde{\mathbf{T}}^{t-1}_{ij} \right\|_2^2, \tag{6.2}$$

where $\mathbf{W}^{t}$, $\mathbf{W}^{t-1} \in \mathbb{R}^{T \times T}$ is a confidence matrix computed for time t and $t-1$ respectively. $\odot$ is element-wise multiplication. For $\mathbf{W}^{t}$, $\mathbf{W}^{t-1}$, we use the visibility and the incident angle of the ray from the camera center to each texel (texture pixel in UV space) as a confidence, enabling the down-weighting of texture distortion caused at grazing angles. For elements smaller than a threshold, they are set to 0. $W^{t,t-1}$ is the number of non-zero elements in $\mathbf{W}^{t} \odot \mathbf{W}^{t-1}$. The intermediate results to compute this

loss are shown in Figure 6.3.

$\widetilde{\mathbf{T}}$ is obtained by projecting the 3D location of each texel decoded from $\mathbf{z}$ to an observed image $\mathbf{I}$, as

$$\widetilde{\mathbf{T}}_{ij} = \mathbf{I}(\mathbf{\Pi}\mathbf{H}_\mathbf{I}\mathbf{X}(\mathbf{G}_\mathbf{I}, i, j)), \tag{6.3}$$

where $\mathbf{H}_\mathbf{I}$ is the rigid head pose regressed from $\mathbf{I}$ via I2ZNet. $\mathbf{G}_\mathbf{I}$ is the set of vertex locations decoded from $\mathrm{D}(\mathbf{z}_\mathbf{I})$, where $\mathbf{z}_\mathbf{I}$ is regressed from $\mathbf{I}$ via I2ZNet. $\mathbf{X}(\cdot)$ computes the 3D location of texel (i, j) in UV space when projected onto $\mathbf{G}_\mathbf{I}$. $\mathbf{\Pi}$ is a weak perspective projection matrix parameterized with a focal length f of the hand-held camera (*i.e.*, $\mathbf{\Pi} = [f\ 0\ 0; 0\ f\ 0]$) that we solve during the domain adaptation. Note that, unlike existing methods that compute per-vertex texture loss [139, **?**], L_{CFTC} considers all visible texels, thus providing significantly richer source of supervision and gradients than per-vertex-based methods. The aforementioned steps are all differentiable, thus the entire model can be updated end-to-end.

Model-to-Observation Texture Consistency

This loss enforces the predicted textures $\mathbf{T}$ to match the texture observed in the image $\widetilde{\mathbf{T}}$. Although this is similar to the photometric loss used in [139] a challenge in our technique is the aforementioned domain mismatch: $\mathbf{T}$ could be significantly different from $\widetilde{\mathbf{T}}$ mainly due to lighting condition changes. Therefore, we incorporate an additional network $\mathbf{T} \leftarrow \mathrm{C}(\mathbf{T})$ to convert the predicted texture to the currently observed texture. $\mathrm{C}(\mathbf{T})$ is also learned, and since training data is limited, we learn a single 1-by-1 convolutional filter which can be viewed as the color correction matrix and corrects the white-balance between the two textures. The model-to-observation texture consistency (MOTC) is formulated as

$$L^t_{\mathrm{MOTC}} = \frac{1}{W^t} \sum_{i,j} \mathbf{W}^t_{ij} \left\| \widetilde{\mathbf{T}}^t_{ij} - \mathrm{C}\left(\mathbf{T}^t_{ij}\right) \right\|_2^2. \tag{6.4}$$

W^t denotes to the number of non-zero elements in $\mathbf{W}^t$.

Facial Landmark Reprojection Consistency

This loss enforces a sparse set of vertices on the 3D mesh corresponding to the landmark locations to be consistent with 2D landmark predictions. This is similar to the landmark

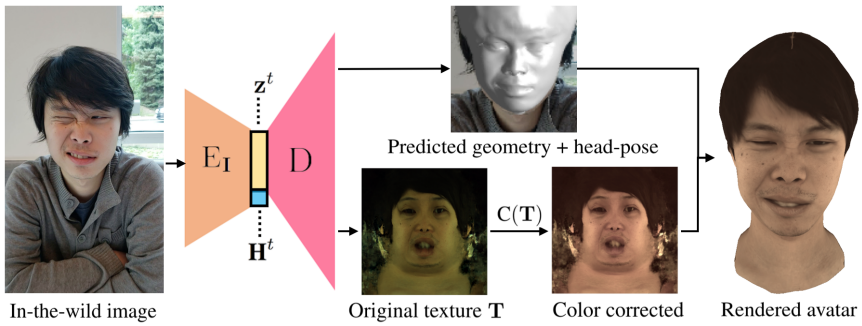


Figure 6.4. Proposed method during testing phase.

reprojection loss used in [139, 253]. Given K facial landmarks, the facial landmark reprojection consistency (FLRC) loss is formulated as

$$L^t_{\mathrm{FLRC}} = \frac{1}{K}\sum_{i=1}^{K} \left\| \mathbf{K}^t_{i,2\mathrm{D}} - \mathbf{\Pi H_I K}^t_{i,\mathbf{G_I}} \right\|_2^2, \tag{6.5}$$

where $\mathbf{K}^t_{i,2\mathrm{D}}$ is the location of i-th landmark detected from the image, and $\mathbf{K}^t_{i,\mathbf{G_I}}$ is the 3D location of the vertex corresponding to i-th landmark that is computed from predicted $\mathbf{G_I}$.

6.1.1 Testing Phase

Figure 6.4 depicts the steps required during the testing phase of our network, which is simply a feed-forward pass through the already adapted $\mathrm{E_I}$ and the already estimated color correction function C. Note that computing $\widetilde{\mathbf{T}}_{ij}$ with Eq. 6.3 and running landmark detection with K are no longer required. Therefore, the timing of the network is still exactly the same as the original network except for the additional color correction, which itself is very simple thus very fast. In sum, our proposed domain adaptation only affects network training time and *does not affect network prediction time.*

6.2 Validation

To demonstrate the effectiveness of our proposed self-supervised domain adaptation method for high-fidelity 3D face tracking, we perform both quantitative and qualitative

analysis. Though qualitative analysis is relatively straight forward, quantitative analysis for evaluating the accuracy and stability of tracking results requires a high-resolution in-the-wild video dataset with ground-truth 3D meshes, which unfortunately is difficult to collect because scanning high quality 3D facial scans usually requires being in a special lab environment with controlled settings. Thus quantitative analysis of recent 3D face tracking methods such as [139, 134] are limited to static image datasets [133], or video sequences shot in a controlled environment [257]. Therefore, in light of the aforementioned limitations, we collected a new dataset and devised two metrics for quantitatively evaluating 3D face tracking performance.

Evaluation Metrics We employ two metrics, accuracy and temporal stability, which are denoted as "Reprojection" and "Temporal" in Table 6.1, respectively. For accuracy, since we do not have ground truth 3D meshes for in-the-wild data, we utilize average 2D landmark reprojection error as a proxy for the accuracy of the predicted 3D geometry. First, a 3D point corresponding to a 2D landmark is projected into 2D, and then the Euclidean distance between the reprojected point and ground truth 2D point is computed. For temporal stability, we propose a smoothness metric as

$$\frac{1}{G}\sum_{i=1}^{G}\frac{\left\|\mathbf{G}_i^{t+1}-\mathbf{G}_i^{t}\right\|_2+\left\|\mathbf{G}_i^{t}-\mathbf{G}_i^{t-1}\right\|_2}{\left\|\mathbf{G}_i^{t+1}-\mathbf{G}_i^{t-1}\right\|_2}, \tag{6.6}$$

where $\mathbf{G}_i^t$ corresponds to the 3D location of vertex i at time t. This metric assumes that the vertices of the 3D mesh should move on a straight line over the course of three frames, thus unstable or jittering predictions will lead to higher (worse) score. The lowest (best) metric score is 1.

Dataset Collection and Annotation We recorded 1920×1080 resolution facial performance data in the wild for four different identities. Recording environments include indoor, outdoor, plain background and cluttered background under various lighting conditions.

150 frames of facial performance data were annotated for each of the 4 identities. For each frame, we annotate on the person's face 5 salient landmarks that do **not** correspond to any typical facial landmark such as eye corners and mouth corners that can be detected by our landmark detector. These points are selected because our domain

Table 6.1. Evaluation on in-the-wild dataset. "Ours w/o DA" represents $\mathbf{E_I}$ before doing any domain adaptation.

		Subject1	Subject2	Subject3	Subject4	Average
HPEN	Temporal	1.5197	1.2951	1.8206	1.3559	1.4978
	Reprojection	8.8075	**5.5475**	13.3823	10.4688	9.5515
3DDFA	Temporal	1.5503	1.4500	1.8608	1.5139	1.5938
	Reprojection	14.1171	10.2568	21.5077	18.1647	16.011
PRNet	Temporal	1.5551	1.3701	1.5700	1.4973	1.4981
	Reprojection	8.4867	7.2522	14.052	9.6586	9.8624
Ours w/o DA	Temporal	1.4106	1.2476	1.8322	1.4169	1.4768
	Reprojection	6.2171	7.4914	10.9225	9.5953	8.5566
Ours w/ L_{FLRC}	Temporal	1.3624	1.3274	1.6583	1.132	1.3700
	Reprojection	5.7558	6.982	10.1258	7.5230	7.5960
Ours	Temporal	**1.1299**	**1.0498**	**1.2934**	**1.0915**	**1.1412**
	Reprojection	**5.5689**	6.7281	**9.6015**	**7.1368**	**7.2588**

adaptation method already optimizes for facial landmark reprojection consistency, so our evaluation metric should use a separate set of landmarks for evaluation. Therefore, we focus on annotating salient personalized landmarks, such as pimples or moles on a person's face, which can be easily identified and accurately annotated by a human. In this way, our annotations enable us to measure performance of tracking in regions where there are no generic facial landmarks and provide a more accurate measure of tracking performance.

Implementation Details : DAMs [60] are first created for all four identities from multi-view images captured in a lighting-controlled environment, and our I2ZNet is newly trained for each identity. Our proposed self-supervised domain adaptation method is then applied to videos of the four identities in a different lighting and background environment. For DAM, the unwrapped texture resolution is $T = 1024$, and the geometry had $G = 7306$ vertices. We train the I2ZNet with Stochastic Gradient Decent

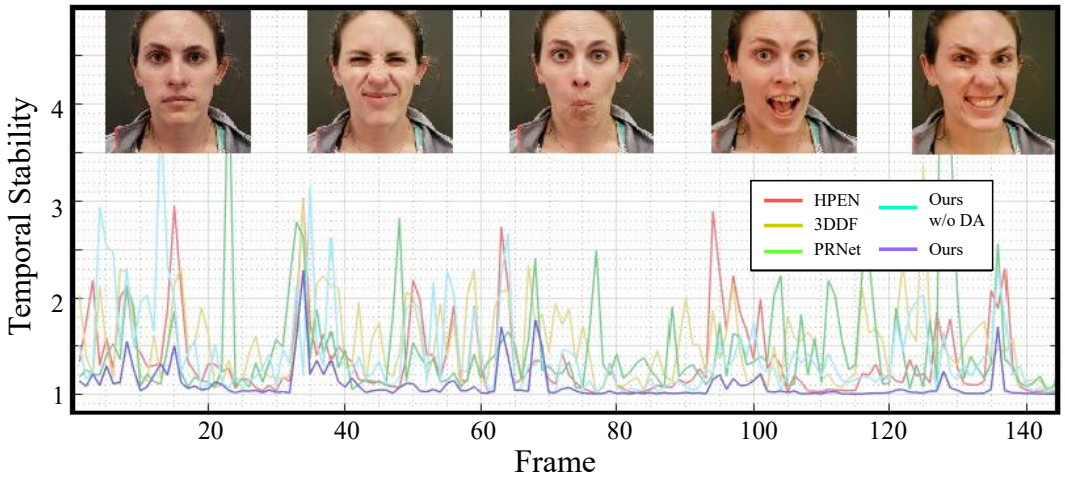


Figure 6.5. Temporal stability graph for subject 4. Note that smaller stability score means more stable results.

(SGD). The face is cropped and resized to 256×256 image and given to $E_{\mathbf{I}}$. During the self-supervised domain adaptation, the related parameters are set to $\lambda_{\text{CFTC}} = 100,\ \lambda_{\text{MOTC}} = 100,\ \lambda_{\text{FLRC}} = 1$.

6.2.1 Results on In-the-wild Dataset

We compare our method against three state-of-the-art baselines: **HPEN** [**?**]: 3DMM fitting based on landmarks, **3DDFA** [81]: 3DMM fitting based on landmarks and dense correspondence, and **PRNet** [140]: 3DMM fitting based on the direct depth regression map. The system input image size is 256×256 except for **3DDFA** (100×100). We also add our method without domain adaptation (**Ours w/o DA**) and only with facial landmark reprojection consistency (**Ours w/** $L_{\mathbf{FLRC}}$). As shown in Table 6.1, the proposed domain adaptation consistently increases the performance of the our model without domain adaptation for all 4 subjects. In terms of stability, the proposed domain adaptation method improves our model by 22% relative. Particularly, we are able to achieve 1.05 stability score for subject 2, which is close to the lowest possible stability score (1.0). This demonstrates the effectiveness of our proposed method. For the other baselines, our model without the domain adaptation already outperforms them in terms of geometry. This may be because our model is pre-trained with many pairs of $(\mathbf{I}, \mathbf{H}, \mathbf{z})$ training data, while the baselines were used out of the box. But on the other hand, all

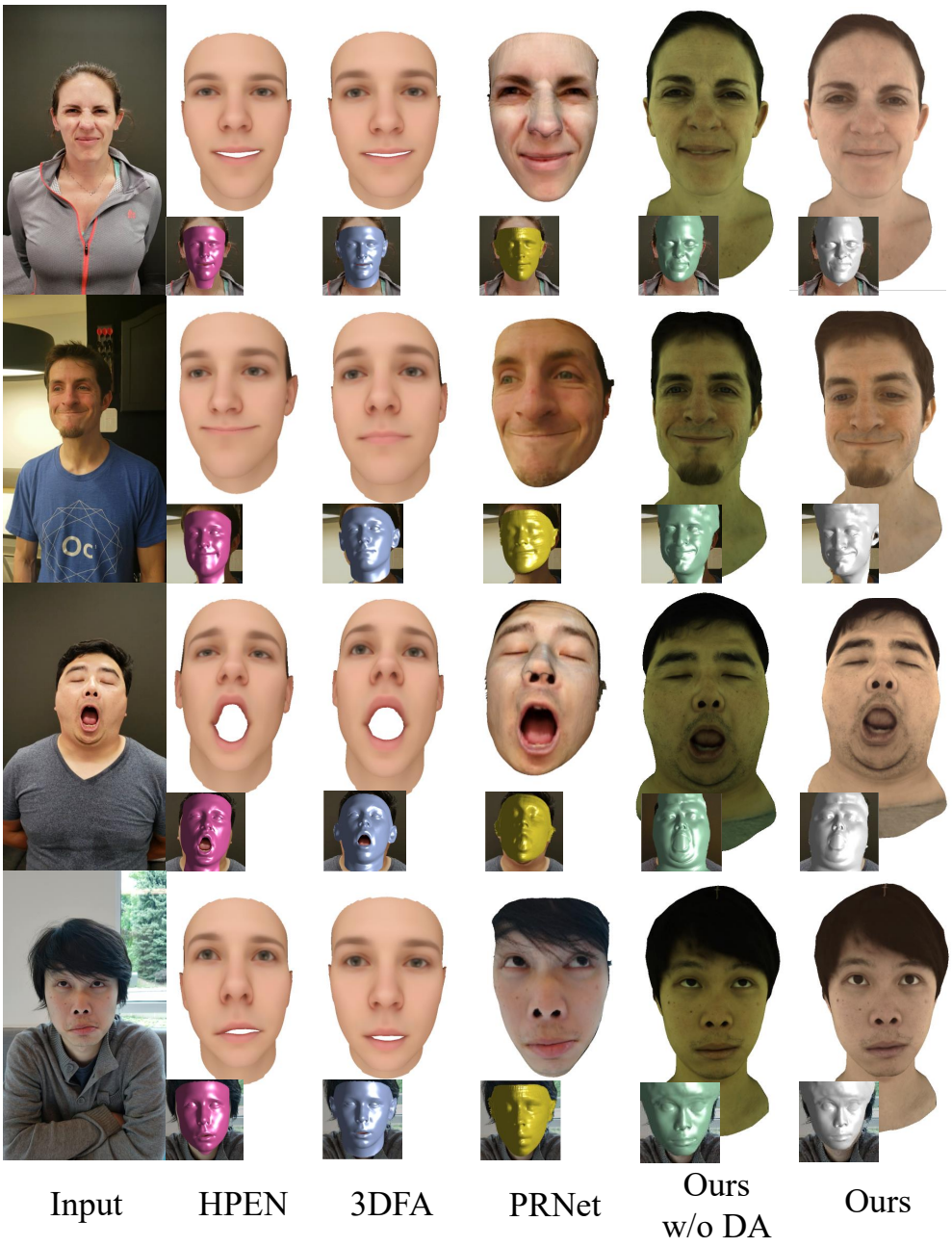


Figure 6.6. Qualitative comparisons with baseline methods.

baselines including **Ours w/o DA** perform similarly in terms of stability (between 1.45-1.60), but our domain adaptation method is able to improve it to 1.14, which clearly demonstrates the effectiveness of our method.

Figure 6.5 visualizes the temporal stability metric for all the different methods for a single sequence. Our method has a consistently better (*i.e.*, smaller) stability score than all the other methods for nearly all the frames, and demonstrates not only the effectiveness, but also the reliability and robustness of our method for in-the-wild sequences.

Figure 6.6 shows qualitative comparisons with baselines. Overall, our face tracking results most closely resemble the input facial configuration, especially for the eyes and the mouth. For example, in the second row, the baselines erroneously predicted that the person's mouth is opened, while our method correctly predicted that the person's mouth is closed. We can also clearly see the effectiveness of our color correction approach, which is able to correct the relatively green-looking face to better match to the appearance in the input.

Figure 6.7 shows the visualization of our in-the-wild face tracking results. Our method is able to track complex motion in many different backgrounds, head pose, and lighting conditions that are difficult to approximate with spherical harmonics such as hard shadow. Our method is also able to adapt to the white-balance of the current scene. Note that the gaze direction is also tracked for most cases.

6.2.2 Effect of Image Resolution

The cropped image resolution plays a key role in the accuracy of face tracking. In this experiment, we quantify the performance degradation according to the resolution using relative reprojection error metric. Relative reprojection error is computed by comparing the 2D reprojected vertices location of the estimated geometry from different resolution images with the one of the gold-standard geometry, which is the geometry acquired when using the highest image resolution 256×256. Figure 6.8 shows the results. Until 175×175, we achieve average error less than 4 pixel-error, but performance degrades significantly as the resolution becomes further smaller.

Figure 6.7. Visualization of 3D face tracking for in-the-wild video. For each input image, we show in the bottom right corner the predicted geometry overlaid on top of the face, and the predicted color corrected face.

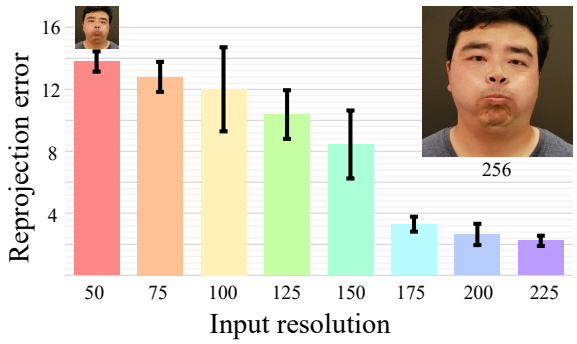


Figure 6.8. Ablation studies on the performance degradation under various input resolution.

6.2.3 Limitations

There are two main limitations to the proposed approach. The first limitation is that our method assumes that a person-specific DAM already exists for the person to be tracked, as our method takes the DAM as input. The second limitation is that our MOTC color correction cannot handle complex lighting and specularities. For example, in Figure 6.7 first row first image, a portion of the face is brighter due to the sun, but since we only have a global color correction matrix for color correction, the sun's effect could not be captured and thus not reflected in the output.

6.3 Summary

We present a self-supervised domain adaptation method to adapt a 3D face model to monocular imagery from new domains, thus enabling high-fidelity face performance tracking to be applied to in-the-wild data. Our method leverages the assumption that the texture of a face over two consecutive frames should not change drastically, and this assumption enables us to extract supervision from unlabeled in-the-wild video frames to fine-tune the existing face tracker and perform self-supervised domain adaptation. The key strength of this approach is that we do not make any other assumptions on the scene or lighting of in-the-wild imagery, enabling our method to be applicable to a wide variety of scenes. The results demonstrate that our proposed method not only improves face-tracking accuracy, but also the stability of tracking.

Chapter 7

Self-Supervised Depth Estimation for Novel View Synthesis of Dynamic Scenes

Novel view synthesis of human requires the reconstruction of the underlying 3D geometries. To obtain them, existing methods [258, 259, 260] have utilized AI models that can predict the depth from a single image by learning monocular cues such as perspective, relative size, occultation, and texture gradient. However, such cues are not consistent with respect to camera viewpoints, which affects the way the AIs perceive how far a person appears to be from the camera. This leads to geometrically incorrect novel view synthesis results as shown in Figure 7.1-(e).

We address this challenge by leveraging the following complementary visual and motion cues: (1) Multi-view images can be combined to reconstruct incomplete yet view-invariant static scene geometry[1], which enables synthesizing a novel view image of static contents in a geometrically consistent way as shown in Figure 7.1-(c). (2) Relative depth predicted from a single image provides view-variant [261] yet complete dynamic scene geometry, which allows enforcing locally consistent 3D scene flow for the foreground dynamic contents.

[1]Its fixed scale chosen from SfM pipeline is consistent across different views from initial triangulation [235].

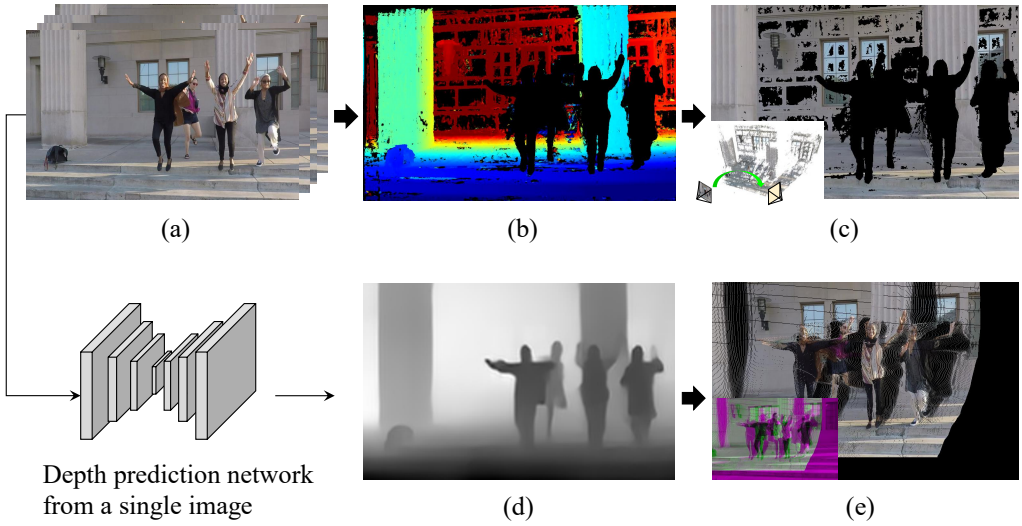


Figure 7.1. Comparison of multi-view and single-view depth estimation. (a) Input images where the background scene is static and foreground (humans) are dynamic. That is, the local and body poses of people are time-varying. (b) Depth estimation from the set of multi-view images by using existing multiview stereo (DMV) approach [21]. (c) Novel view synthesis with DMV which produces incomplete yet geometrically correct rendering results. (d) Depth estimation from each single image by using existing single-view depth prediction (DSV) method [22]. (e) Novel view synthesis with DSV which produces complete yet geometrically incorrect results. The overlay with the ground-truth is shown as inset.

We combine these cues by learning a nonlinear scale correction function that can upgrade a time series of single view geometries to form a coherent 4D reconstruction. To disambiguate the geometry of the foreground dynamic contents, we find their simplest motion description in 3D (i.e., slow and smooth motion [262, 263]), which generates minimal stereoscopic disparity when seen by a novel view [264].

We model the scale correction function that takes input images, view-variant depth from single view (DSV), and incomplete yet view-invariant depth from a multi-view stereo (DMV) algorithm, and outputs complete and view-invariant depth. The network is self-supervised by three visual signals without any labeled data: (i) the static regions of the DSV must be aligned with a DMV; (ii) the output depth of dynamic regions must be consistent with the relative depth of each DSV; and (iii) the estimated scene flow must be minimal and locally consistent. With the predicted depths that are geometrically

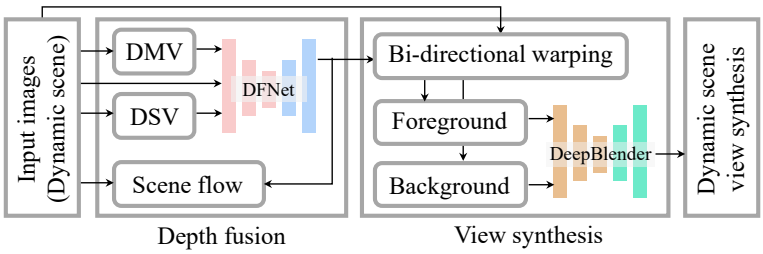


Figure 7.2. Images of a dynamic scene are used to predict and estimate the depth from single view (DSV) and the depth from multi-view stereo (DMV). Our depth fusion network (DFNet) fuses the individual strengths of DSV and DMV (Sec. 7.1.1) to produce a complete and view-invariant depth by enforcing geometric consistency. The computed depth is used to synthesize a novel view and our DeepBlnder network refines the synthesized image (Sec. 7.1.2).

consistent across views, we synthesize a novel view using a self-supervised rendering network that produces a photorealistic image in the presence of missing data with adversarial training. An overview of our pipeline is shown in Figure 7.2.

7.1 Approach

We cast the novel view synthesis problem as image warping from input source views to a virtual view using underlying 4D reconstruction, i.e.,

$$\mathbf{J}^v(W_{r\to v}(\mathbf{x})) = \mathbf{I}^r(\mathbf{x}), \tag{7.1}$$

where $\mathbf{J}^v$ is the synthesized image from an arbitrary virtual view v (v can be a source viewpoint), $W_{r\to v}$ is a warping function, and $\mathbf{I}^r$ is the r^{th} source image.

For view synthesis of static scene, the warping function can be described as:

$$\mathbf{y} = W_{r\to v}(\mathbf{x}; \mathbf{D}^r, \Pi^r, \Pi^v), \tag{7.2}$$

where Π^r and Π^v are the projection matrices at the r^{th} and v^{th} viewpoints. The warping function forms the warped coordinates $\mathbf{y}$ by reconstructing the view-invariant 3D geometry using the depth ($\mathbf{D}^r$) and projection matrix at the r^{th} viewpoint, and projecting onto the v^{th} viewpoint. For instance, this warping function can generate the i^{th} source image from the j^{th} source image, i.e., $\mathbf{I}^i(W_{j\to i}) = \mathbf{I}^j$.

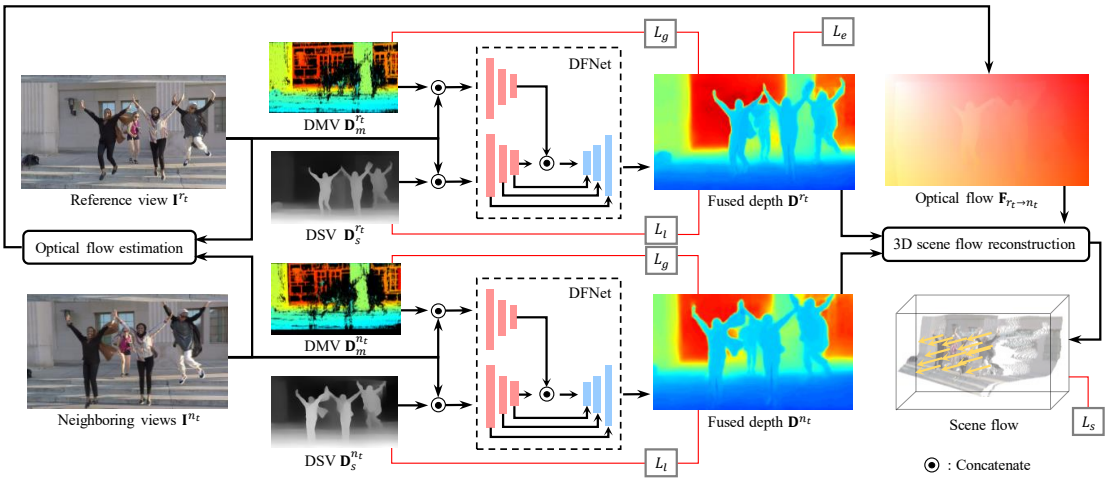


Figure 7.3. Depth Fusion Network (DFNet) predicts a complete and view-invariant depth map by fusing DSV and DMV with the image. DFNet is self-supervised by minimizing the background depth consistency with DMV (L_g), the relative depth consistency with DSV (L_l), 3D scene flow (L_s), and spatial irregularity (L_e).

For view synthesis of dynamic scene, the warping function can be generalized to include the time-varying geometry using the depth $\mathbf{D}^{r_t}$, i.e.,

$$\mathbf{y} = W_{r_t \to v}(\mathbf{x}; \mathbf{D}^{r_t}, \Pi^r, \Pi^v), \tag{7.3}$$

where r_t is the time dependent view index, and t is the time instant. Note that for a moving monocular camera, the view is a function of time. Unlike the static scene warping $W_{r \to v}$ in Equation (7.2), we cannot synthesize i^{th} source image from the j^{th} source image because of the time-varying geometry $\mathbf{D}^{r_t}$, i.e., $\mathbf{I}^i(W_{j \to i}) \neq \mathbf{I}^j$.

With these two warping functions, the dynamic scene view synthesis can be expressed as:

$$\mathbf{J} = \phi\left(\left\{\mathbf{J}^v\left(W_{r \to v}\right)\right\}_r, \mathbf{J}^{v,t}\left(W_{r_t \to v}\right); \mathcal{M}^v\right), \tag{7.4}$$

where $\{\mathbf{J}^v(W_{r \to v})\}_r$ is a set of static scene warping from all source viewpoints, and $\mathbf{J}^{v,t}(W_{r_t \to v,t})$ is the warping of dynamic contents from the source image of the t^{th} time instant. $\mathcal{M}^v$ is the set of the coordinates belonging to dynamic contents. ϕ is the rendering function that refines the warped images to complete the view synthesis.

In Equation (7.4), two quantities are unknowns: the depth from each source view $\mathbf{D}^{r_t}$ and the rendering function ϕ. We formulate these two quantities in Sec. 7.1.1 and Sec. 7.1.2.

7.1.1 Globally Coherent Depth from Dynamic Scenes

Our conjecture is that there exists a scale correction function that can upgrade a complete view-variant depth $\mathbf{D}_s^{r_t}$ from the single view prediction (DSV) to the depth of the view-invariant 3D geometry $\widehat{\mathbf{D}}^{r_t}$:

$$\widehat{\mathbf{D}}^{r_t} = \psi(\mathbf{D}_s^{r_t}), \tag{7.5}$$

where ψ is the scale correction function. Ideally, when a scene is stationary, the upgraded depth is expected to be identical to the depth $\mathbf{D}_m^r$ from view-invariant geometry, e.g., depth from multiview stereo (DMV), with uniform scaling, i.e., $\mathbf{D}_m^r = \psi(\mathbf{D}_s) = \alpha\mathbf{D}_s + \beta$ where α and β are scalar and bias. When a scene is dynamic, the linear regression of such scale and bias is not applicable. We learn a nonlinear scale correction function that possesses the following three properties.

First, for the static scene, the upgraded depth approximates DMV:

$$\mathbf{D}_m^r(\mathbf{x}) \approx \psi\left(\mathbf{D}_s^{r_t}(\mathbf{x})\right) \quad \text{for } \mathbf{x} \notin \mathcal{M}^{r_t}, \tag{7.6}$$

where $\mathbf{x}$ is the coordinate of pixels belonging to the static background.

Second, for the dynamic contents, the upgraded depth preserves the relative depth from DSV:

$$g\left(\mathbf{D}_s^{r_t}(\mathbf{x})\right) \approx g\left(\psi\left(\mathbf{D}_s^{r_t}(\mathbf{x})\right)\right) \quad \text{for } \mathbf{x} \in \mathcal{M}^{r_t}, \tag{7.7}$$

where g measures the scale invariant relative gradient of depth, i.e.,

$$g(\mathbf{D}; \mathbf{x}, \Delta\mathbf{x}) = \frac{\mathbf{D}(\mathbf{x}+\Delta\mathbf{x}) - \mathbf{D}(\mathbf{x})}{|\mathbf{D}(\mathbf{x}+\Delta\mathbf{x})| + |\mathbf{D}(\mathbf{x})|}. \tag{7.8}$$

We use multi-scale neighbors $\mathbf{x} + \Delta\mathbf{x}$ to constrain local and global relative gradients.

Third, 3D scene motion induced by the upgraded depths is smooth and slow [59], i.e., minimal scene flow:

$$\mathbf{p}(\mathbf{x}; \mathbf{D}^{r_t}, \Pi^{r_t}) \approx \mathbf{p}(F_{r_t \to n_t}(\mathbf{x}); \mathbf{D}^{n_t}, \Pi^{n_t}), \tag{7.9}$$

where $F_{r_t \to n_t}$ is the optical flow from the r_t^{th} to n_t^{th} source images. $\mathbf{p}(\mathbf{x}; \mathbf{D}) \in R^3$ is the reconstructed point in the world coordinate using the depth $\mathbf{D}$:

$$\mathbf{p}(\mathbf{x}; \mathbf{D}, \Pi) = \psi\left(\mathbf{D}(\mathbf{x})\right) \mathbf{R}^{\mathsf{T}} \mathbf{K}^{-1} \widetilde{\mathbf{x}} + \mathbf{C} \tag{7.10}$$

where $\widetilde{\mathbf{x}}$ is the homogeneous representation of $\mathbf{x}$, and $\mathbf{R} \in SO(3)$, $\mathbf{C} \in R^3$, and $\mathbf{K}$ are the camera rotation matrix, camera optical center, and camera intrinsic parameters from the projection matrix Π.

Depth Fusion Network (DFNet) We enable the scale correction function ψ using a depth fusion network that takes as input DSV, DMV, and image $\mathbf{I}^{r_t}$:

$$\widehat{\mathbf{D}}^{r_t} = \psi\left(\mathbf{D}_s^{r_t}, \mathbf{D}_m^{r_t}, \mathbf{I}^{r_t}; \mathbf{w}\right), \tag{7.11}$$

where the network is parametrized by its weights $\mathbf{w}$. To learn $\mathbf{w}$, we minimize the following loss:

$$L(\mathbf{w}) = L_g + \lambda_l L_l + \lambda_s L_s + \lambda_e L_e, \tag{7.12}$$

where λ controls the importance of each loss. L_g measures the difference between DMV and the estimated depth in Equation (7.6) for static scene:

$$L_g = \|\widehat{\mathbf{D}}^{r_t}(\mathbf{x}) - \mathbf{D}_m^{r_t}(\mathbf{x})\| \quad \text{for} \quad \mathbf{x} \notin \mathcal{M}^{r_t},$$

L_l compares the scale invariant depth gradient between DSV and the estimated depth in Equation (7.7):

$$L_l = \|g(\widehat{\mathbf{D}}^{r_t}(\mathbf{x})) - g(\mathbf{D}_s^{r_t})(\mathbf{x})\| \quad \text{for} \quad \mathbf{x} \in \mathcal{M}^{r_t},$$

and L_s minimize the induced 3D scene motion for entire pixel coordinates in Equation (7.9):

$$L_s = \|\mathbf{p}(\mathbf{x}; \mathbf{D}^{r_t}, \Pi^{r_t}) - \mathbf{p}(F_{r_t \to n_t}(\mathbf{x}); \mathbf{D}^{n_t}, \Pi^{n_t})\|.$$

In conjunction with self-supervision, we further minimize the Laplacian of the estimated depth as regularization, i.e.,

$$L_e = \|\nabla^2 \widehat{\mathbf{D}}^{r_t}(\mathbf{x})\|^2 + \lambda_f \|\nabla^2 \widehat{\mathbf{D}}^{r_t}(\bar{\mathbf{x}})\|^2 \tag{7.13}$$

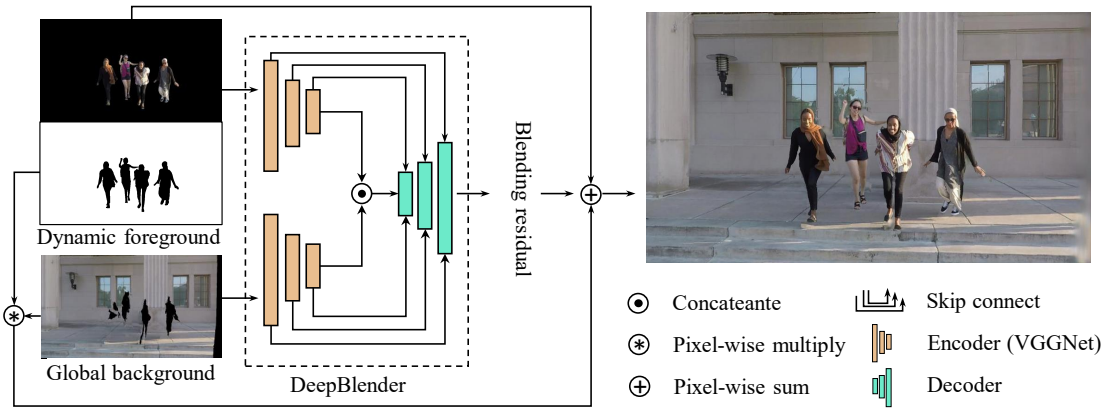


Figure 7.4. View synthesis pipeline: Given the warped foreground (FG) and background (BG) through the depths and masks, we complete the dynamic scene view synthesis using a rendering network called DeepBlender that predicts the missing region and refines the artifacts.

where $\mathbf{x} \notin \mathcal{M}^{r_t}$, $\bar{\mathbf{x}} \in \mathcal{M}^{r_t}$, and λ_f balances the spatial smoothness between the static and dynamic regions.

The overview of our self-supervision pipeline and the network architecture are described in Figure 7.3. DFNet extracts the visual features from DSV and DMV using the same encoder in conjunction with the image. With the visual features, DFNet generates a complte and view invariant depth map that is geometrically consistent. To preserve the local visual features, skip connections between the feature extractor and depth generator are used.

7.1.2 Dynamic Scene View Synthesis

Given a set of warped static scenes from all source views $\{\mathbf{J}^v\}_r$, we construct a global background $\mathbf{J}^v_*$ based on the baseline between the virtual and source cameras, i.e., assign the pixel value from the warped source view that has the shortest baseline with virtual camera. With $\mathbf{J}^v_*$ and the warped dynamic contents $\mathbf{J}^{v,t}$ from a single time instant, we model the synthesis function ϕ in Equation (7.4) as follows:

$$\phi(\mathbf{J}^v_*, \mathbf{J}^{v,t}; \mathcal{M}^v) = \mathbf{J}^v_*(\mathbf{x}) + \mathbf{J}^{v,t}(\mathbf{y}) + \widetilde{\phi}_\theta(\mathbf{J}^v_*, \mathbf{J}^{v,t}), \tag{7.14}$$

where $\mathbf{x} \notin \mathcal{M}^{v,t}$ and $\mathbf{y} \in \mathcal{M}^{v,t}$. $\widetilde{\phi}_\theta$ is the blending residual that fills the missing regions (unlike a static scene, there exists the regions that are not seen by any source views for

a dynamic scene) and refines the synthesized image. We model this blending residual $\widetilde{\phi}_\theta$ using our rendering network.

DeepBlender Network The DeepBlender predicts the blending residual $\widetilde{\phi}_\theta$ from the inputs of a warped dynamic scene $\mathbf{J}^{v,t}$ and a globally modeled static scene $\mathbf{J}^v_*$ as shown in Figure 7.4. It combines visual features extracted from $\mathbf{J}^{v,t}$ and $\mathbf{J}^v_*$ to form a decoder with skip connections. We learn this rendering function using source images with self-supervision. Each image is segmented into background and foreground with the corresponding foreground mask. We synthetically generate the missing regions near the foreground boundary and image border, and random pixel noises across the scenes. From the foreground and background images with missing regions and pixel noises, the DeepBlender is trained to generate the in-painting residuals. We incorporate an adversarial loss to produce photorealistic image synthesis:

$$L(\mathbf{w}_\theta) = L_{\text{rec}} + \lambda_{\text{adv}} L_{\text{adv}}, \tag{7.15}$$

where L_{rec} is the reconstruction loss (difference between the estimated blending residual and ground truth), and L_{adv} is the adversarial loss [265]. The overview of our view synthesis pipeline is described in Figure 7.4.

7.2 Implementation Details

DFNet is pre-trained on a synthetic dataset [266] (which provides ground-truth optical flow, depth, and foreground mask) for better weight initialization during the self-supervision. To simulate the characteristic of the real data from synthetic, we partially remove the depth around the foreground region and add the depth noise across the scenes with 5% tolerance of the variance at every training iteration. The same self-supervision loss as Equation 7.15 is used to pre-train the network. To avoid the network depth scale confusion, we use the normalized inverse depth [22] for both DMV and DSV and recover the scale of the fused depth based on the original scale of DMV. To obtain DSV and DMV, we use existing single view prediction [22] and multiview stereo method [21]. In Equation (7.8), we use five multi-scale neighbors, i.e., $\Delta\mathbf{x} = \{1, 2, 4, 8, 16\}$ to consider both local and global regions. We use PWCNet [267] to compute the optical flow

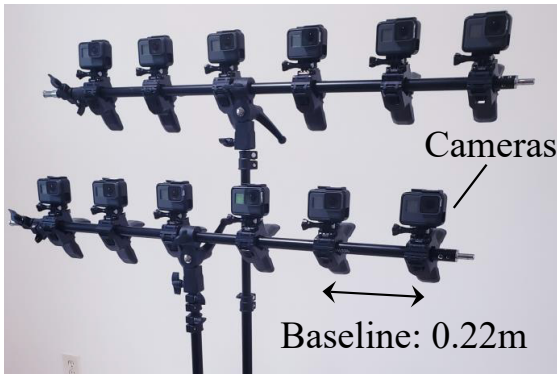


Figure 7.5. Camera rig.

in Equation (7.9), where the outliers were handled by forward-backward flow consistency. When enforcing the scene flow loss, we use ± 2 neighboring camera views, i.e., $n_t = r_t \pm 2$. We extract the foreground mask using interactive segmentation tools [241]. The foreground masks are manually specified for all baselines in the evaluation, while existing foreground segmentation approaches [268] can be used as a complementary tool as shown in Figure 7.8.

We also pre-train the DeepBlender using video object segmentation dataset [269]. To create the synthetic residual, we randomly generate the seams and holes around the foreground using mask morphology and superpixel, and remove one side of the image boundary up to 30-pixel thickness. The loss in Equation 7.15 is used for pre-training as well. When we warp an image to a virtual view, we check bidirectional warping consistency to prevent the pixel holes. For each image warping, we refine the depth using the bilateral weighted median filters [270]. As shown in Figure 7.4, we handle the foreground and background separately to prevent the pixel mixing problem around the object boundary.

7.3 Experiments

We evaluate our method with various dynamic scenes.

F+B / F-only ↘	Jumping	Skating	Truck	DynaFace	Umbrella	Balloon1	Balloon2	Teadybear	Avg.
MVS [21]	0.53 / 2.12	0.29 / 6.81	0.52 / 2.94	0.05 / 0.21	**0.35** / 4.70	0.13 / 1.72	**0.04** / 0.31	**0.06** / 0.92	0.24 / 2.46
RMVSNet [271]	0.61 / 1.55	0.76 / 1.56	0.84 / 2.43	2.24 / 1.57	0.67 / 5.24	0.23 / 1.40	0.13 / 0.38	0.58 / 0.89	0.75 / 1.87
MonoDepth [22]	1.79 / 2.55	1.34 / 2.02	2.62 / 3.86	0.39 / 0.74	2.69 / 4.75	1.07 / 1.88	1.06 / 0.99	0.76 / 0.28	1.46 / 2.13
Sparse2Dense [272]	1.35 / 3.26	1.35 / 10.66	2.15 / 7.60	0.20 / 0.34	1.35 / 6.40	0.53 / 3.03	0.48 / 0.65	0.32 / 0.90	0.96 / 4.10
DFNet-L_g	1.26 / 1.31	0.81 / 0.76	1.60 / 1.24	0.26 / 0.91	2.19 / 1.98	0.93 / 1.36	0.53 / 0.30	1.91 / 0.97	1.18 / 1.10
DFNet-L_l	0.46 / 1.58	0.15 / 1.38	0.62 / 3.34	0.09 / 0.26	0.58 / 3.14	0.15 / 1.57	0.08 / 0.30	0.16 / 0.67	0.28 / 1.53
DFNet-L_e	0.38 / 0.93	0.14 / 0.47	0.52 / 1.09	0.07 / 0.12	0.52 / 2.48	0.15 / 1.20	0.06 / 0.24	0.17 / 0.48	0.26 / 0.87
DFNet-L_s	0.37 / 1.09	0.14 / 0.51	0.53 / 1.11	0.07 / 0.13	0.59 / 2.54	0.16 / 1.18	0.07 / 0.25	0.16 / 0.52	0.26 / 0.91
DFNet	**0.35** / **0.76**	**0.12** / **0.40**	**0.41** / **0.83**	**0.03** / **0.08**	0.37 / **1.90**	**0.12** / **1.11**	0.05 / **0.23**	0.17 / **0.32**	**0.20** / **0.70**

Table 7.1. Results of quantitative evaluation for the task of depth estimation from dynamic scenes. RMSE in the metric scale is used for evaluation. F and B represent the foreground and background, respectively. The lower is the better.

7.3.1 Dynamic Scene Dataset

We collect dynamic scenes using two methods. (1) Moving monocular camera: short-term dynamic events ($\sim$ 5s) are captured by a hand-held monocular moving camera (Samsung Galaxy Note 10) with 60Hz framerate and 1920×1080 resolution. We sub-sample the sequence if the object motion is not salient, and therefore, the degree of the scene motion is significantly larger than that of the camera egomotion where quasi-static dynamic reconstruction does not apply. Four dynamic scenes are captured, which

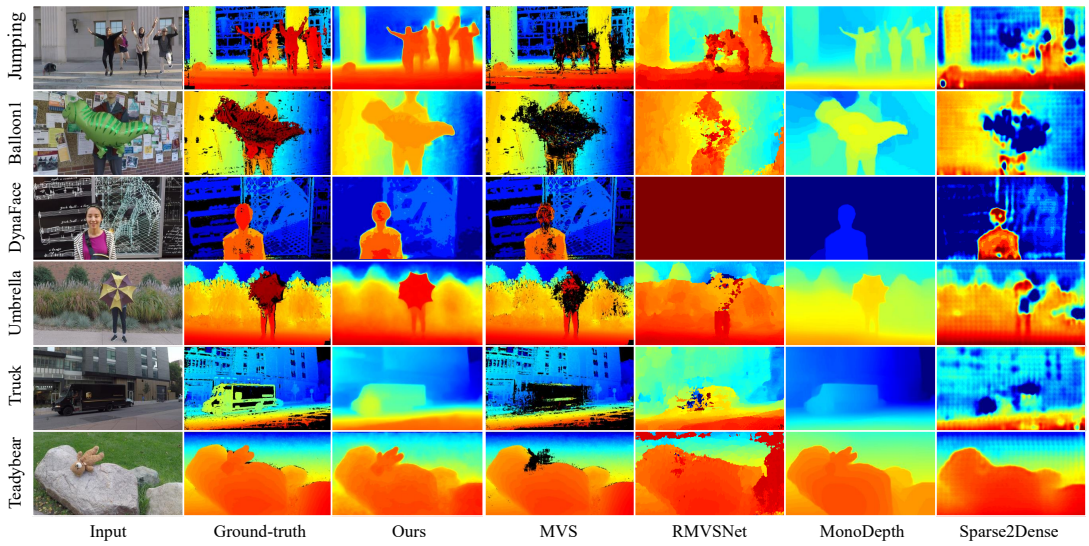


Figure 7.6. Qualitative comparison of the dynamic scene depth estimation from each method.

includes human activity, human-object interaction, and animal movement (see the supplementary video). These scenes are used for the qualitative evaluation, where we use half-resolution inputs. (2) Stationary multiview cameras: 8 scenes are captured by a static camera rig with 12 cameras (GoPro Black Edition), where the ground truth of depth estimation and view synthesis are available for the quantitative evaluation. The cameras are located at two levels, and at each level, 6 cameras are evenly distributed with 0.22m baseline as shown in Figure 7.5. All cameras are manually synchronized. The dataset is categorized into following: (1) Human: a single or multiple people show their dynamic motion, e.g., dynamic facial expression and body motion. (2) Interaction: a person interacts with objects, e.g., umbrella, balloon, and skate. (3) Vehicle: a truck rigidly move from the right side of the road to the left. (4) Stop motion: a doll is sequentially captured in the different location. When testing, we use a set of images sampled from each camera at different time instant to simulate a moving monocular camera. Given the set of collected images, we calibrate the intrinsic and extrinsic parameters of the moving camera using structure-from-motion [76].

7.3.2 Quantitative Evaluation Metric

We evaluate the accuracy of depth estimation and view synthesis using the multiview dataset. (1) Depth estimation: given the estimated depth, we measure root mean square error (RMSE) by comparing to the ground-truth depth computed by multiview stereo. The error is represented in metric scale (m), i.e., the scale of the estimated depth is upgraded to the metric space using the physical length of the camera baseline. We exclude the region that cannot be reconstructed by multiview stereo. (2) View synthesis: we measure the mean of the optical flow magnitude from the ground-truth image to the synthesized one to validate the view invariant property of the depth map. Ideally, it should be close to 0 with the perfect depth map. Additionally, we measure the perceptual similarity [63] (i.e., the distance of VGG features) with the ground-truth to evaluate the visual plausibility of the synthesized view, where its range is normalized into [0, 1] (the lower is the better).

7.3.3 Baselines and Ablation Study

We compare our depth estimation and view synthesis methods with a set of baseline approaches. For the depth evaluation, we compare our method with four baselines: 1) Multiview stereo (MVS [21]) assumes that a scene is stationary. For the pixel of which MVS failed to measure the depth, we assign the average of valid depth. 2) RMVSNet [271] is a learning based multiview stereo algorithm. 3) MonoDepth [22] predicts the depth from a single view image. As it produces the normalized depth, we re-scale the predicted depth by using the mean and standard deviation from MVS depth. 4) Sparse2Dense [272] completes the depth given an incomplete depth estimation, where we use MVS depth as an input. As this method requires the metric depth, we upgrade the estimated depth to the metric space using the physical length of the camera baseline. In conjunction with comparative evaluations, we conduct an ablation study to validate the choice of losses.

For the view synthesis evaluation, we compare our view warping method (bi-directional 3D warping) with as-similar-as-possible warping [273] which warps an image by estimating grid-wise affine transforms. The correspondences of the warping are computed by projecting the estimated depth, i.e., transporting pixels in a source image to a novel

view through the view-invariant depth. In Table 7.2, we denote bi-directional warping followed by the DeepBlender refinement as B3W, and as-similar-as-possible warping followed by the DeepBlender as ASAPW. Note that the DeepBlender refinement is applied to all methods except for DFNet+B3W-DeepBlender which evaluates the effect of the refinement by eliminating the DeepBlender. On top of the comparison with different warping methods, we also test all possible combination of depth estimation methods with view warping methods as listed in Table 7.2. It quantifies how the quality of depth maps affect the view synthesis results.

7.3.4 Dynamic Scene Depth Estimation

In Table 7.1, we summarize the accuracy of dynamic scene depth estimation results evaluated on: 1) the entire scene, and 2) the only dynamic contents. For the entire scene, our method shows the best results on average, followed by MVS with 0.04 m accuracy gap. In the sequence of umbrella and teadybear, MVS shows the better accuracy for the entire scene than ours due to the highly occupant background area as shown in Figure 7.6, i.e., the depth estimation of dynamic contents much less contributes to depth accuracy evaluation than one of the background. From the evaluation on the only dynamic contents, our method (DFNet) also shows the best result with the noticeable accuracy improvement (1.17 m) from the second best method (MonoDepth).

While the relative depth of MonoDepth is well reflective of the ground-truth, its depth range is often biased to a specific range, e.g., the foreground object is located much closer to the background scenes. Sparse2Dense does not fully reconstruct the background depth even with the MVS depth as inputs, and the predicted foreground depth is completely incorrect. It indicates that fusing the individual strength of learning-base and stereo-based geometry is essential to obtain the globally coherent and complete depth map from dynamic scenes. From Figure 7.6, we can further notice that the learning based multiview stereo (RMVSNet) also fail to model the dynamic foreground geometry. In our experiment, RMVSNet completely fail when the object is too close to the camera.

From the ablation study described in Table 7.1, L_g is the most critical self-supervision signal as the MVS depth plays the key role to convey the accurate static depth. Those accurate depths play the fiducial point for the other self-supervision signals to predict

the depth on the missing area. From DFNet-L_l, we can verify that the single view depth estimation can upgrade the depth accuracy around the dynamic contents by guiding it with accurate relative depths. Although the contribution of L_e and L_s are relatively small than others, it helps to regularize the object scene motion and the spatial smoothness of the foreground depth which are keys to reduce the artifacts of the novel view synthesis.

7.3.5 Dynamic Scene Novel View Synthesis

Table 7.2 shows the quantitative evaluation of view synthesis, and the associated qualitative results are shown in Figure 7.7. From the qualitative results, we can notice that two types of artifacts can be produced depending on the warping methods: B3W produces flying pixel noises, i.e., a pixel is floating due to the warping with incorrect depths, while ASAPW produces image distortion. Such artifacts lead to the increase of the perceptual distance with ground-truths as it captures the structural similarity. On average, our method (DFNet+B3W) shows the smallest perceptual distance (0.15), indicating that the geometry from our depth map is highly preservative of scene structure. The comparison of DFNet+B3W with DFNet+ASAPW demonstrates that, given an accurate depth map, pixel-wise warping (B3W) is the better choice over the grid-wise warping (ASAWP) for view synthesis. From the results of DFNet+B3W-Deepblender, we can observe the large improvement of perceptual similarity compared to the results without DeepBlender, indicating that the refinement step (hole filling and noise reduction) is essential for visual plausibility.

Our method (DFNet+B3W) performs the best even for the flow evaluation (5.3 pixels). MVS+B3W is following our method with the 6.8 pixel errors but it produces a significant pixel noise around the dynamic contents as shown in Figure 7.7. While MonoDepth+B3W reconstructs visually plausible results in Figure 7.7, it accompany with large flow errors (10.8 pixels on average), meaning that this result is not geometrically plausible. Note that, the optical flow error of DFNet+B3W-Deepblender is much higher than DFNet+B3W because the flow estimation algorithm [267] shows significant confusion when there are holes around the image boundary and dynamic contents.

Limitation It is worth noting a few limitations of our method. The DFNet may not perform well when a viewing angle between neighboring views are larger (e.g., rotating

Perceptual Sim. / Optical Flow ↘	Jumping	Skating	Truck	DynaFace	Umbrella	Balloon1	Balloon2	Teadybear	Avg.
MVS [21]+ASAPW [273]	0.21 / 7.0	0.17 / 9.3	0.10 / 4.0	0.30 / 19.0	0.19 / 7.5	0.23 / 16.0	0.17 / 6.7	1.80 / 4.9	0.19 / 9.3
RMVSNet [271]+ASAPW [273]	0.22 / 6.4	0.23 / 13.1	0.11 / 3.4	0.98 / 10.2	0.19 / 7.2	0.23 / 14.9	0.16 / 6.3	0.20 / 10.0	0.29 / 8.9
MonoDepth [22]+ASAPW [273]	0.23 / 9.1	0.18 / 11.8	0.10 / 5.1	0.32 / 20.9	0.20 / 9.8	0.25 / 17.3	0.23 / 11.4	0.17 / 7.8	0.20 / 11.7
Sparse2Dense [272]+ASAPW [273]	0.23 / 7.5	0.19 / 9.4	0.11 / 4.8	0.31 / 20.8	0.19 / 7.0	0.23 / 13.7	0.16 / 6.6	0.19 / 6.4	0.20 / 9.52
MVS [21]+B3W	0.24 / 7.0	0.20 / 9.2	0.12 / 3.5	0.27 / 7.5	0.19 / 5.7	0.23 / 14.4	0.17 / 5.4	0.13 / **1.5**	0.19 / 6.8
RMVSNet [271]+B3W	0.23 / 5.6	0.23 / 14.8	0.14 / 3.3	1.0 / 10.8	0.19 / 5.6	0.23 / 12.0	0.16 / 5.1	0.19 / 8.9	0.29 / 8.2
MonoDepth [22]+B3W	0.23 / 8.5	0.18 / 11.4	0.10 / 5.0	0.32 / 19.1	0.19 / 8.5	0.24 / 17.3	0.23 / 11.4	0.15 / 5.2	0.20 / 10.8
Sparse2Dense [272]+B3W	0.24 / 7.3	0.20 / 9.2	0.13 / 4.7	0.31 / 11.7	0.2 / 6.7	0.24 / 14.0	0.18 / 6.6	0.17 / 4.8	0.22 / 8.12
DFNet+ASAPW [273]	0.20 / 5.8	0.17 / 9.3	0.09 / 3.0	0.30 / 18.0	0.18 / 6.4	0.20 / 13.3	0.16 / 6.4	0.17 / 5.8	0.18 / 8.5
DFNet+B3W-DeepBlender	0.23 / 8.2	0.21 / 13.1	0.12 / 4.8	0.30 / 15.6	0.22 / 9.0	0.25 / 15.8	0.20 / 9.2	0.18 / 4.7	0.21 / 10.1
DFNet+B3W (**ours**)	**0.16** / **4.2**	**0.15** / **8.8**	**0.08** / **2.5**	**0.22** / **6.2**	**0.16** / **3.6**	**0.18** / **10.6**	**0.14** / **5.1**	**0.13** / 2.0	**0.15** / **5.3**

Table 7.2. Quantitative evaluation results on the dynamic scene novel view synthesis task. To measure the accuracy, we compute perceptual similarity and optical flow magnitude between the ground-truth and the synthesized image.

more than 45°), which may decrease the amount of overlaps of dynamic contents. If the scene is highly cluttered by many objects from both background and foreground (e.g., many people, thin poles, and trees), our pipeline could cause noisy warping results due to the significant depth discontinuities from the clutter. Our method will fail in the scenes where the camera calibration does not work, e.g., a scene largely occupied by dynamic contents [266]. Finally, our view synthesis with completely failed foreground mask produces significant artifacts such as afterimages and object fragmentation as shown in Figure 7.8.

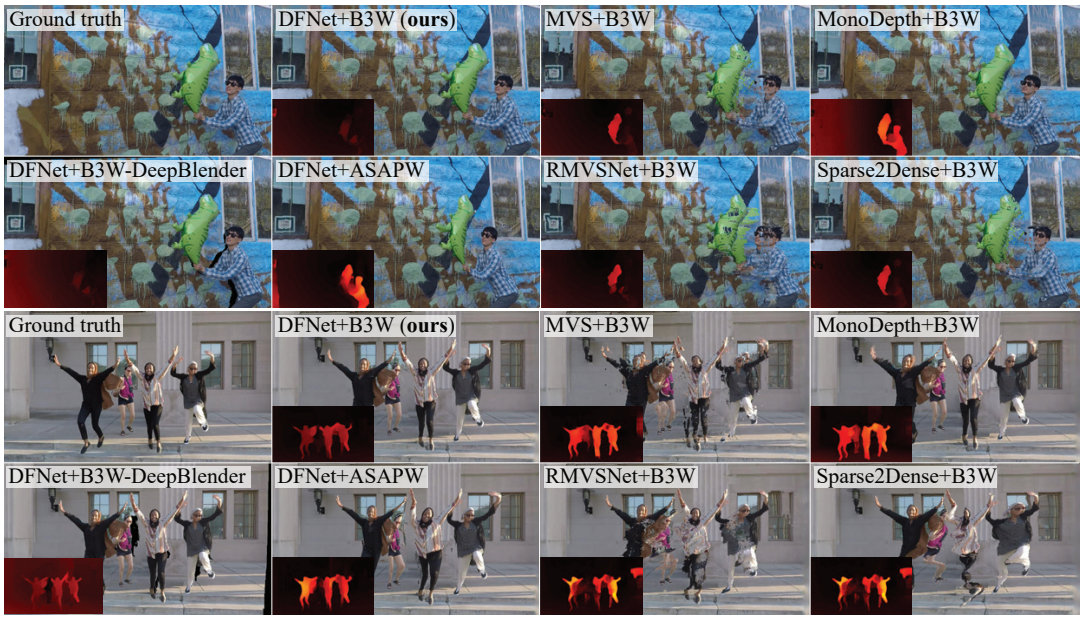


Figure 7.7. Qualitative comparison on the view synthesis task. The pixel error is shown in the inset image (maximum pixel error is set to 50 RGB distance).

7.4 Summary

In this chapter, we study the problem of monocular view synthesis of a dynamic scene including human. The main challenge is to reconstruct dynamic contents to produce geometrically coherent view synthesis, which is an ill-posed problem in general. To address this challenge, we propose to learn a scale correction function that can upgrade the depth from single view (DSV), which allows matching to the depth of the multi-view solution (DMV) for static contents while producing locally consistent scene motion. Given the computed depth, we synthesize a novel view image using the DeepBlender network that is designed to combine foreground, background, and missing regions. Through the evaluations for depth estimation and novel view synthesis, we demonstrate that the proposed method can apply to the daily scenario captured from a monocular camera.

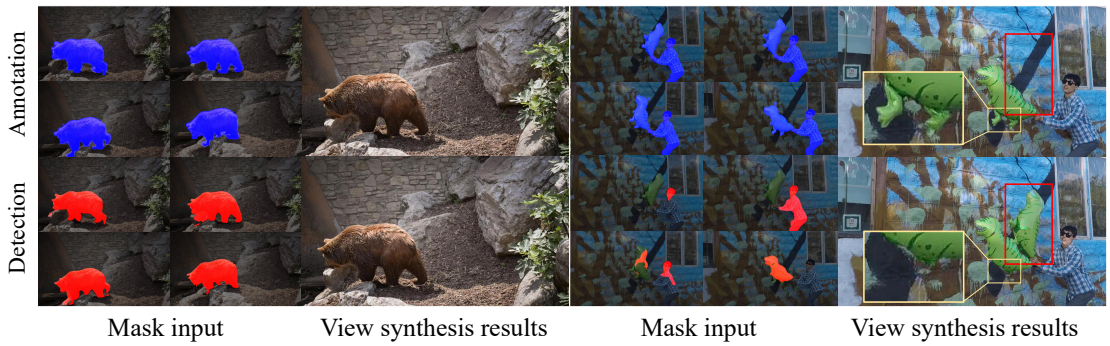


Figure 7.8. The mask detection with small mistakes (left) does not have a significant impact on the view synthesis results. However, if the mask detection is completely failed (right), it produces artifacts such as object fragmentation (yellow box) or afterimage (red box).

Part III

Learning to Render Fine-Grained Appearance of 3D Avatars of Diverse People

Chapter 8

Pose-Guided Human Animation from a Single Image in the Wild

Being able to animate a human in everyday apparel with an arbitrary pose sequence from just a single still image opens the door to many creative applications. For example, animated photographs can be much more memorable than static images. Furthermore, such techniques not only simplify and democratize computer animation for non-experts, they can also expedite pre-visualization and content creation for more professional animators who may use single image animations as basis for further refinement.

Tackling this problem using classical computer graphics techniques is highly complex and time consuming. A high-quality 3D textured human model needs to be reconstructed from a single image and then sophisticated rigging methods are required to obtain an animatable character. An alternative is to apply 2D character animation methods [274, 275] to animate the person in the image. However, this approach cannot visualize the occluded parts of the character.

In this chapter, we approach this problem using a pose transfer algorithm that synthesizes the appearance of a person at arbitrary pose by transforming the appearance from an input image without requiring a 3D animatable textured human model. Existing works on pose transfer have demonstrated promising results only when training and testing take place on the same dataset (*e.g.,* DeepFashion dataset [1]), and some require even more restrictive conditions that testing is performed on the same person in the same

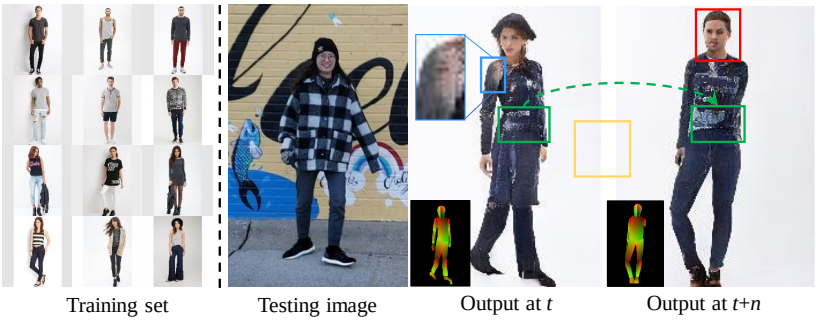


Figure 8.1. The pose transfer results synthesized by a state-of-the-art method [18] on an unconstrained real-world scene, where the network is trained on the Deep Fashion dataset [1]. The target body pose is shown in the inset (black). Each box represents the type of the observed artifacts such as loss of identity (red), misclassified body parts (blue), background mismatch (yellow), and temporal incoherence (green).

environment as training. [26, 203, 207]. However, the domain difference between training and testing data in real applications introduces substantial quality degradation.

A core challenge of pose transfer lies in lack of data that span diverse poses, shapes, appearance, viewpoints, and background. This leads to limited generalizability to a testing scene, resulting in noticeable visual artifacts as shown in Fig. 8.1. We address this challenge by decomposing the pose transfer task into modular subtasks predicting silhouette, garment labels, and textures where each task can be learned from a large amount of synthetic data. This modularized design makes training tractable and significantly improves result quality. Explicit silhouette prediction further facilitates animation blending with arbitrary static scene backgrounds.

In inference phase, given the trained network from the synthetic data, we introduce an efficient strategy for synthesizing temporally coherent human animations controlled by a sequence of body poses. We first produce a unified representation of appearance and its labels in UV coordinates, which remains constant across different poses. This unified representation provides an incomplete yet strong guidance to generating the appearance in response to the pose change. We use the trained network to complete the appearance and render it with the background. Experiments show that our method significantly outperforms the state-of-the-art methods in terms of synthesis quality, temporal consistency, and generalization ability.

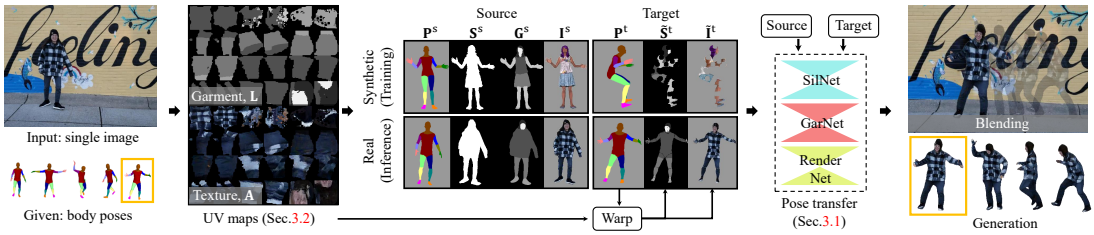


Figure 8.2. Overview of our approach. Given an image of a person and a sequence of body poses, we aim for generating video-realistic human animation. To this end, we train a compositional pose transfer network that predicts silhouette, garment labels, and textures with synthetic data (Sec. 8.1.1). In inference phase, we first produce a unified representation of appearance and garment labels in the UV maps, which remains constant across different poses, and these UV maps are conditioned on our pose transfer network to generate person images in a temporally consistent way (Sec. 8.1.6). The generated images are composited with the inpainted background to produce the animation.

8.1 Methodology

Our goal is to synthesize human animations from a single image guided with a sequence of arbitrary body poses. The overview of our pipeline is outlined in Fig. 8.2. In the training stage, our pose transfer network learns to generate a person's appearance in different poses using a synthetic dataset which provides full ground truth. At inference time, given a single image of a person and a different body pose, the learned pose transfer network generates the person's appearance that is conditioned on the partial garment and texture warped from the coherent UV maps (scene-specific priors). The generated foreground is blended with the inpainted background. In Sec. 8.1.1, we introduce our compositional pose transfer network, and in inference time, we use this network to create coherent UV maps and human animation from a single image in Sec. 8.1.6.

8.1.1 Compositional Pose Transfer

The problem of pose transfer takes as input a source image $\mathbf{I}^{\mathrm{s}}$ and a target pose $\mathbf{P}^{\mathrm{t}}$ and generates an image of the person in the target pose $\mathbf{I}^{\mathrm{t}}$:

$$\mathbf{I}^{\mathrm{t}} = f(\mathbf{P}^{\mathrm{t}}, \mathbf{I}^{\mathrm{s}}). \tag{8.1}$$

where the superscript s denotes the source as the domain of the observation from the input image, and t denotes the target as of the generation from a body pose.

Albeit possible, directly learning the function in Eq. (8.1) is challenging as requiring large amount of multiview data [179, 188, 18], *i.e.,* it requires to learn the deformation of the shape and appearance with respect to every possible 3D pose, view, and clothing style. This results in a synthesis of unrealistic human images that are not reflective of the input testing image as shown in Fig. 8.1. We address this challenge by leveraging synthetic data that allows us to decompose the the function into the modular functions that are responsible to predict silhouette, garment labels, and appearance, respectively. This makes the learning task tractable and adaptable to the input testing image.

8.1.2 Dataset and Notation

For training, we use 3D people synthetic dataset [2] which contains 80 subjects in diverse clothing styles with 70 actions per subject captured from four different virtual views, where each action is a sequence of 3D poses. For each subject we randomly pick two instances as the source and target with different views and 3D poses. Each instance contains the following associated information:

- Image: $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ is the person image where the foreground is masked using $\mathbf{S}$.
- Pose map: $\mathbf{P} \in \{0, \cdots, 14\}^{W \times H}$ is a map of body-part labels of the undressed body (14 body parts and background).
- Silhouette mask: $\mathbf{S} \in \{0, 1\}^{W \times H}$ is a binary map indicating one if it belongs to the person foreground, and zero otherwise.
- Garment labels: $\mathbf{G} \in \{0, \cdots, 6\}^{W \times H}$ is a map of garment labels of dressed human body, indicating hair, face, skin, shoes, garment top and bottom, and background.

In inference time, given $\mathbf{I}^{\mathrm{s}}$ and $\mathbf{P}^{\mathrm{s}}$, we estimate the $\mathbf{P}^{\mathrm{s}}, \mathbf{S}^{\mathrm{s}}$ and $\mathbf{G}^{\mathrm{s}}$ from $\mathbf{I}^{\mathrm{s}}$ using off-the-shelf methods, and our pose transfer network predicts $\mathbf{S}^{\mathrm{t}}, \mathbf{G}^{\mathrm{t}}$, and $\mathbf{I}^{\mathrm{t}}$.

8.1.3 Silhouette Prediction

We predict the silhouette of the person in the target pose given the input source triplet: source pose map, silhouette, and garment label. It is designed to learn the shape

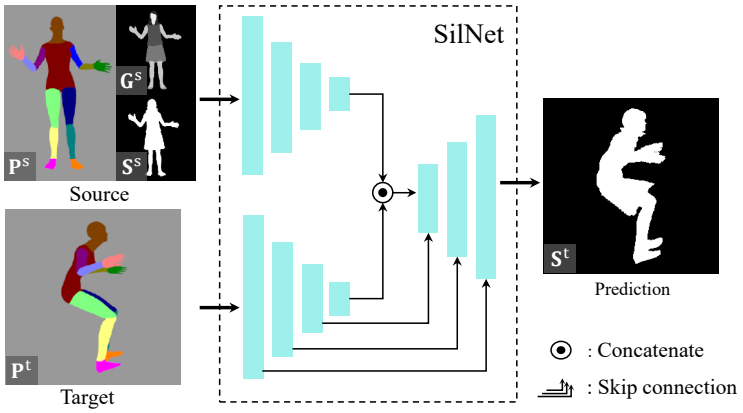


Figure 8.3. *SilNet* predicts the silhouette mask in the target pose.

deformation as a function of the pose change:

$$\mathbf{S}^t = f^{\text{Sil}}(\mathbf{P}^t | \{\mathbf{P}^s, \mathbf{S}^s, \mathbf{G}^s\}). \tag{8.2}$$

We use a neural network called *SilNet* to learn this function. It has two encoders and one decoder, as shown in Fig. 8.3. One encoder encodes the spatial relationship of the body and silhouette from the source triplet, which is used to condition the silhouette generation of the target pose by mixing their latent codes. The garment labels $\mathbf{G}^s$ provides an additional spatial cue to control the deformation, *i.e.,* pixels that do not belong to garment (*i.e.,* skin) less likely undergo large deformation. The features extracted from the target pose at each level are passed to the counterpart of the decoder through skip connections. We train *SilNet* by minimizing the $L1$ distance of the predicted silhouette mask $\mathbf{S}^t$ and the ground truth $\mathbf{S}^t_{\text{gt}}$:

$$L_{\text{Sil}} = \left\| \mathbf{S}^t - \mathbf{S}^t_{\text{gt}} \right\|_1 . \tag{8.3}$$

Note that, as f^{Sil} does not take as input the source image $\mathbf{I}^s$, using synthetic data does not introduce the domain gap.

8.1.4 Garment Label Prediction

Given the source triplet and the predicted target silhouette, we predict the target garment labels $\mathbf{G}^t$ that guide the generation of the target appearance. We take two steps.

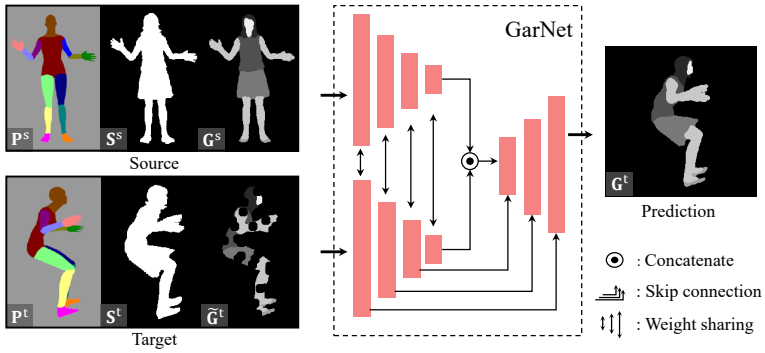


Figure 8.4. *GarNet* predicts the garment labels in the target pose.

First, we warp the source garment labels to produce the pseudo target garment labels, $\widetilde{\mathbf{G}}^{\mathrm{t}}$,

$$\widetilde{\mathbf{G}}^{\mathrm{t}}(\mathbf{x}) = \mathbf{G}^{\mathrm{s}}(\mathcal{W}_{\mathrm{s}}^{-1}(\mathcal{W}_{\mathrm{t}}(\mathbf{x}))), \tag{8.4}$$

where $\mathcal{W}_{\mathrm{s}}, \mathcal{W}_{\mathrm{t}} : R^2 \rightarrow R^2$ are the warping functions that transform a point in the source and target image $\mathbf{x}$ to the UV coordinate of the body. The pseudo target garment label is incomplete because the body silhouette is a subset of the dressed body silhouette. Note that this first step, *i.e.,* producing $\widetilde{\mathbf{G}}^{\mathrm{t}}$ by warping, only applies in the inference time, while in training time, we synthetically create the incomplete pseudo garment labels $\widetilde{\mathbf{G}}^{\mathrm{t}}$ by removing the outside region of the body silhouette from the ground truth $\mathbf{G}^{\mathrm{t}}_{\mathrm{gt}}$ and further removing some parts using random binary patches.

Second, given the input triplet and the predicted target silhouette, we complete the full target garment labels $\mathbf{G}^{\mathrm{t}}$:

$$\mathbf{G}^{\mathrm{t}} = f^{\mathrm{Gar}}(\widetilde{\mathbf{G}}^{\mathrm{t}} | \mathbf{P}^{\mathrm{t}}, \mathbf{S}^{\mathrm{t}}, \{\mathbf{P}^{\mathrm{s}}, \mathbf{S}^{\mathrm{s}}, \mathbf{G}^{\mathrm{s}}\}). \tag{8.5}$$

We design a neural network called *GarNet* to learn the target garment label completion. It consists of a Siamese encoder and a decoder, as shown in Fig. 8.4. The Siamese encoder encodes the spatial relationship from both source and target triplets. A decoder completes the garment labels by classifying every pixel in the target silhouette. Similar to *SilNet*, we use skip connections to facilitate the target feature transform. We train *GarNet* by minimizing the following loss:

$$L_{\mathrm{Gar}} = \left\| \mathbf{G}^{\mathrm{t}} - \mathbf{G}^{\mathrm{t}}_{\mathrm{gt}} \right\|_1. \tag{8.6}$$

f^{Gar} does not take as input the source image $\mathbf{I}^{\text{s}}$ where using synthetic data does not introduce the domain gap.

8.1.5 Foreground Rendering

We synthesize the foreground person image in a target pose given the predicted target garment label and the source image triplet: source image, silhouette, and garment label. Similar to the garment label completion in Sec. 8.1.4, we generate the pseudo target image $\widetilde{\mathbf{I}}^{\text{t}}$ and its silhouette $\widetilde{\mathbf{S}}^{\text{t}}$ using the UV coordinate transformation of $\mathcal{W}_{\text{s}}$ and $\mathcal{W}_{\text{t}}$ in inference time, while synthetically create the incomplete $\widetilde{\mathbf{I}}^{\text{t}}$ and $\widetilde{\mathbf{S}}^{\text{t}}$ from the ground truth $\mathbf{I}^{\text{t}}_{\text{gt}}$ and $\mathbf{S}^{\text{t}}_{\text{gt}}$ in training time.

We learn a function that can render the full target foreground image:

$$\mathbf{I}^{\text{t}} = f^{\text{Render}}(\widetilde{\mathbf{I}}^{\text{t}}, \widetilde{\mathbf{S}}^{\text{t}} | \mathbf{S}^{\text{t}}, \mathbf{G}^{\text{t}}, \{\mathbf{I}^{\text{s}}, \mathbf{S}^{\text{s}}, \mathbf{G}^{\text{s}}\}). \tag{8.7}$$

We design a neural network called *RenderNet* to learn this function. As shown in Fig. 8.5, *RenderNet* encodes the spatial relation $\mathbf{z}^{\text{s}}$ of the source image triplet, and mixes the latent representations from the target. We use two encoders to extract the features of the target garment label $\mathbf{G}^{\text{t}}$ and pseudo target image $\widetilde{\mathbf{I}}^{\text{t}}$ where $\mathbf{S}^{\text{t}}$ and $\widetilde{\mathbf{S}}^{\text{t}}$ are combined with them. We condition these features at each level of the decoder using spatially adaptive normalization blocks [23, 24] to guide the network to be aware of the subject's silhouette, and garment and texture style in the target pose.

We train *RenderNet* by minimizing the following loss:

$$L_{Render} = L_{rec} + \lambda_1 L_{\text{VGG}} + \lambda_2 L_{\text{CX}} + \lambda_3 L_{\text{cAdv}} + \lambda_4 L_{\text{KL}},$$

where the weight λ_i are empirically chosen that all the losses have comparable scale.

Reconstruction Loss. L_{rec} measures the per-pixel errors between the synthesized image $\mathbf{I}^{\text{t}}$ and the ground truth $\mathbf{I}^{\text{t}}_{\text{gt}}$: $L_1 = \left\|\mathbf{I}^{\text{t}} - \mathbf{I}^{\text{t}}_{\text{gt}}\right\|_1$.

VGG Loss. Beyond the low-level constraints in the RGB space, L_{VGG} measures the image similarity in the VGG feature space [276] which is effective in generating natural and smooth person image proven by existing works [188, 182, 224]: $L_{\text{VGG}} = \sum_{i=1}^{4} \left\|\text{VGG}_i(\mathbf{I}^{\text{t}}) - \text{VGG}_i(\mathbf{I}^{\text{t}}_{\text{gt}})\right\|_1$, where $\text{VGG}_i(\cdot)$ maps an image to the activation of the conv-i-2 layer of VGG-16 network [16].

Contextual Loss. L_{CX} measures the similarity of two set of features considering global

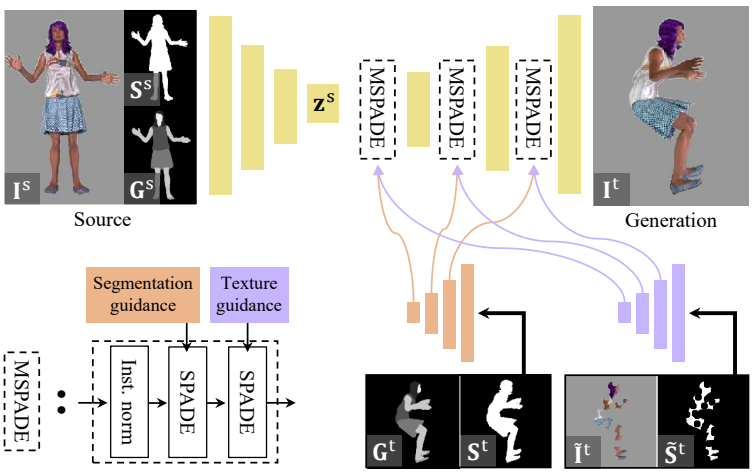


Figure 8.5. *RenderNet* synthesizes the image of a person in the target pose.

image context: $L_{\text{CX}} = -\log(g(\text{VGG}_3(\mathbf{I}^{\text{t}}), \text{VGG}_3(\mathbf{I}^{\text{t}}_{\text{gt}}))$, where $g(\cdot,\cdot) \in [0,1]$ denotes the similarity metric of the matched features based on the normalized cosine distance [277]. Existing work [188] proved that combining L_{CX} with L_{VGG} further helps to preserve the style patterns in the generated image in a semantically meaningful way, *i.e.,* less distorted facial structure.

Adversarial loss. We employ the conditional adversarial loss L_{cAdv} [278] with a discriminator conditioned on garment labels to classify the synthesized image into real or fake, *i.e.,* $\{\mathbf{I}^{\text{t}}_{\text{gt}}, \mathbf{G}^{\text{t}}_{\text{gt}}\}$ is real and $\{\mathbf{I}^{\text{t}}, \mathbf{G}^{t}_{\text{gt}}\}$ is fake. Here, we use the PatchGAN discriminator [279].

KL divergence. L_{KL} is to enforce the latent space $\mathbf{z}^{\text{s}}$ to be close to a standard normal distribution [280, 130].

8.1.6 Consistent Human Animation Creation

With the learned pose transfer network, it is possible to generate the shape and appearance given a target pose map at each time instant. However, it makes independent prediction for each pose, which leads to unrealistic jittery animation. Instead, we build a unified representation of appearance and its labels that provide a consistent guidance across different poses, which enforces the network to predict temporally coherent appearance and shape.

We construct the garment labels $\mathbf{L}$ and textures $\mathbf{A}$ that remain constant in UV

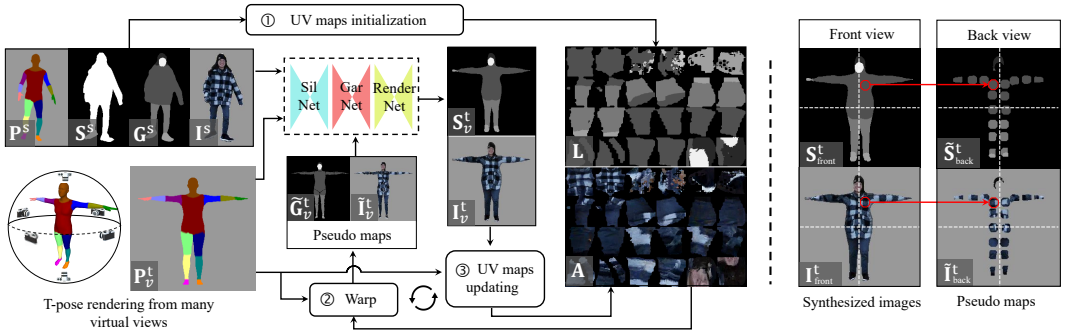


Figure 8.6. We reconstruct the complete UV maps of the garment labels and textures, *i.e.,* $\mathbf{L}$ and $\mathbf{A}$, in an incremental manner. (Left) We first initialize these maps by warping the pixels in the source image, *i.e.,* $\mathbf{I}^{\mathrm{s}}$ and $\mathbf{S}^{\mathrm{s}}$, to the UV maps. We further update the UV maps by combining the synthesized images of a person in a T pose captured from six virtual views. For each virtual view v, we create the pseudo images, *i.e.,* $\widetilde{\mathbf{G}}^{\mathrm{t}}_v$ and $\widetilde{\mathbf{I}}^{\mathrm{t}}_v$, from the previously updated UV maps. (Right) Only for the back view, we construct $\widetilde{\mathbf{G}}^{\mathrm{t}}_v$ and $\widetilde{\mathbf{I}}^{\mathrm{t}}_v$ by sampling the patches from the synthesized images in the frontal view with the front-back symmetry assumption where the face regions are removed.

coordinates by warping the garment label and appearance of an image, *i.e.,* $\mathbf{L}(\mathbf{x}) = \mathbf{G}(\mathcal{W}^{-1}(\mathbf{x}))$ and $\mathbf{A}(\mathbf{x}) = \mathbf{I}(\mathcal{W}^{-1}(\mathbf{x}))$. These UV representations ($\mathbf{L}$ and $\mathbf{A}$) cannot be completed from a single view input image because of occlusion. To complete the UV representations, we use the multiview images synthesized from the rendered 3D human model of which texture is predicted by the learned pose transfer network. This set of generated images are used to incrementally complete the UV representations as shown in Fig. 8.6-(left).

In practice, we generate multiview images by synthesizing the SMPL model at the T pose from six views: front, back, left, right, top and bottom views. We assume that the source image is taken from the frontal view. The back view is generated by applying front-back symmetry assumption [281, 282, 283] as shown in Fig. 8.6-(right).

In the inference phase, this unified UV representation allows us to consistently generate the pseudo garment labels $\widetilde{\mathbf{G}}^{\mathrm{t}}(\mathbf{x}) = \mathbf{L}(\mathcal{W}_{\mathrm{t}}(\mathbf{x}))$ and appearance $\widetilde{\mathbf{I}}^{\mathrm{t}}(\mathbf{x}) = \mathbf{A}(\mathcal{W}_{\mathrm{t}}(\mathbf{x}))$ given a target pose by transforming the SMPL T-pose to the target pose. This pseudo representations provide an incomplete yet strong guidance to the pose transfer network to complete the target foreground.

In order to have both foreground and background in the animation, we segment the foreground from the source image using $\mathbf{S}^{\mathrm{s}}$ and apply an inpainting method [284] to the background. We then composite our synthesized human animation with the background.

8.2 Implementation Details

We train the proposed *SilNet*, *GarNet*, and *RenderNet* separately in a fully supervised way using only 3D people synthetic dataset [2] which is described in Sec 8.1.1. For training, we set the parameters of $\lambda_1 = 0.5$, $\lambda_2 = 0.1$, $\lambda_3 = 0.01$, $\lambda_4 = 10$ and use the Adam optimizer [285] ($lr = 1 \times 10^{-3}$ and $\beta = 0.5$). After training, no further fine-tuning on the testing scene is required. For the pose map $\mathbf{P}$ and garment label map $\mathbf{S}$, we convert them to rgb and gray scale images for the network input.

In inference time, we obtain $\mathbf{S}^{\mathrm{s}}$ and $\mathbf{G}^{\mathrm{s}}$ using person segmentation [244] and fashion segmentation [286]. For $\mathbf{P}^{\mathrm{s}}$, we fit a 3D body model [17] to an image using recent pose estimator [147] and render the parts label onto the image where we follow the same color coding as synthetic data [2]. We generate a sequence of body poses $\{\mathbf{P}_i^{\mathrm{t}}\}_{i=1}^{N}$ by animating the 3D body model using recent motion archive [287], where we represent the z-directional motion as scale variation [153] with weak-perspective camera projection, and rendering the pose map from each body pose similarly to $\mathbf{P}^{\mathrm{s}}$. The image resolution is 256×256, and UV maps are 512×768.

We provide the implementation details of each modular function in our compositional pose transfer network. Fig. 8.8 describes the *SilNet* architecture which takes as input source triplet of the pose map, garment labels, and silhouette, and target pose map, and predicts the silhouette mask in the target pose. Fig. 8.9 describes the architecture of our *GarNet* that takes as input source triplet of the pose map, silhouette, and garment labels, and target triplet of the pose map, predicted silhouette, and pseudo garment labels, and predicts the complete garment labels. In Fig. 8.12, we show the details of our *RenderNet* which takes as input source triplet of image, silhouette mask, and garment labels, target silhouette and garment labels, and target pseudo image and its mask, and generates the person image.

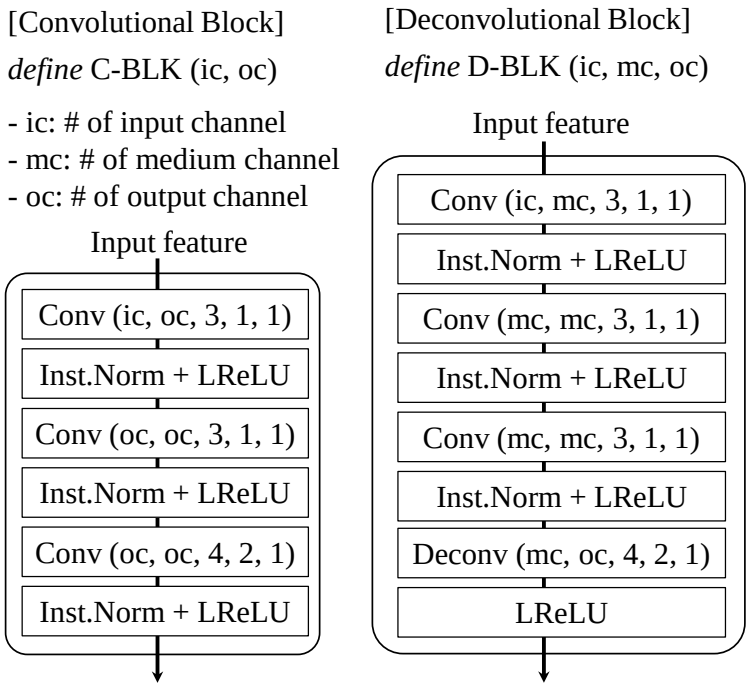


Figure 8.7. Description of our convolutional and deconvolutional blocks. The convolutional (Conv) and deconvolutional layers (Deconv) take parameters including the number of input channels, the number of output channels, filter size, stride, and the size of zero padding. We use 0.2 for the LeakyReLU (LReLU) coefficient.

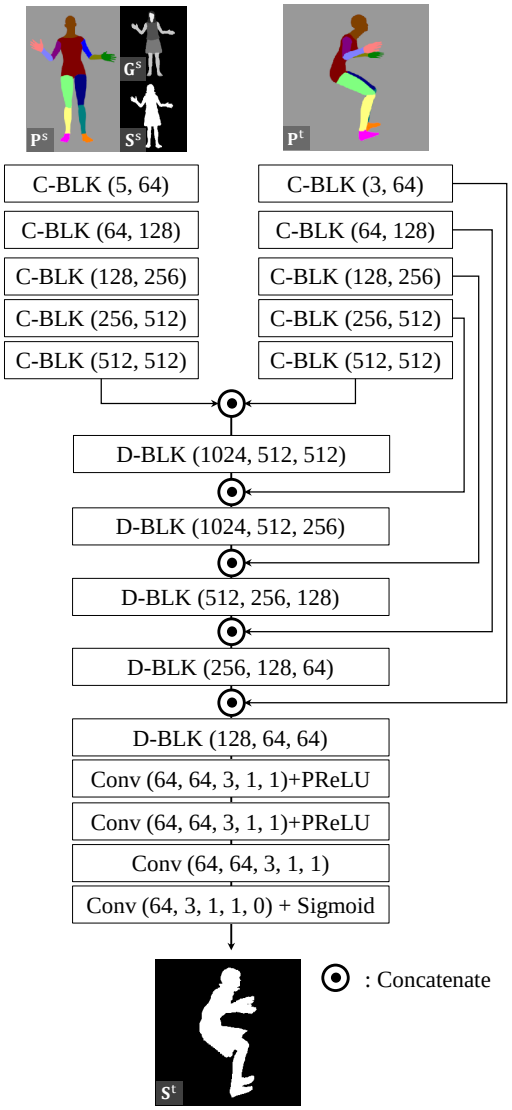


Figure 8.8. The details of our *SilNet* implementation where C-BLK and D-BLK are described in Fig. 8.7. Conv and Deconv take as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.

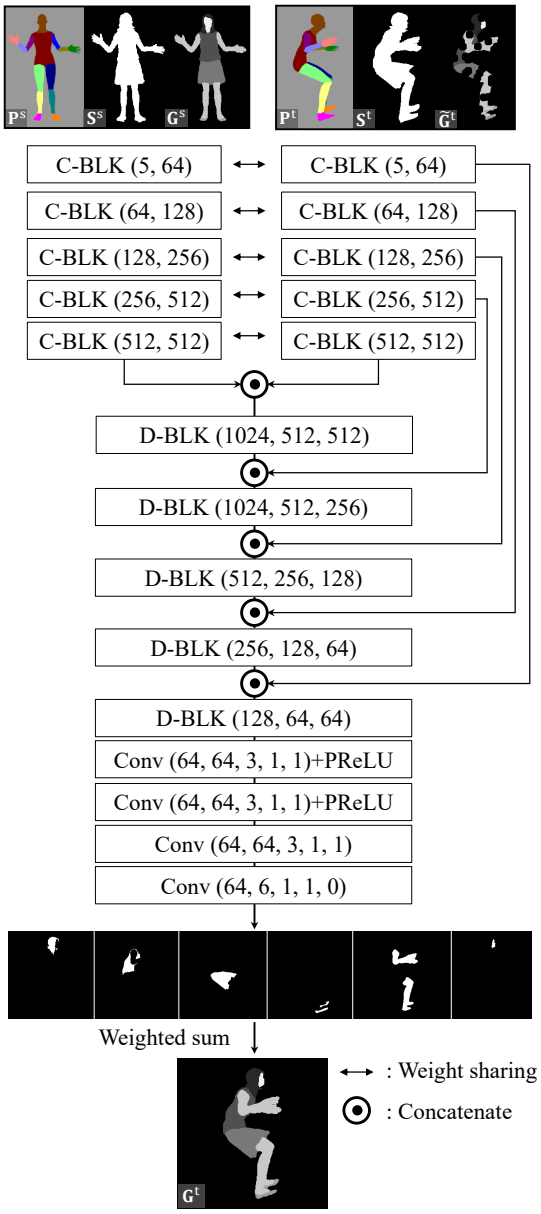


Figure 8.9. The details of our *GarNet* implementation where C-BLK and D-BLK are described in Fig. 8.7. Conv and Deconv take as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.

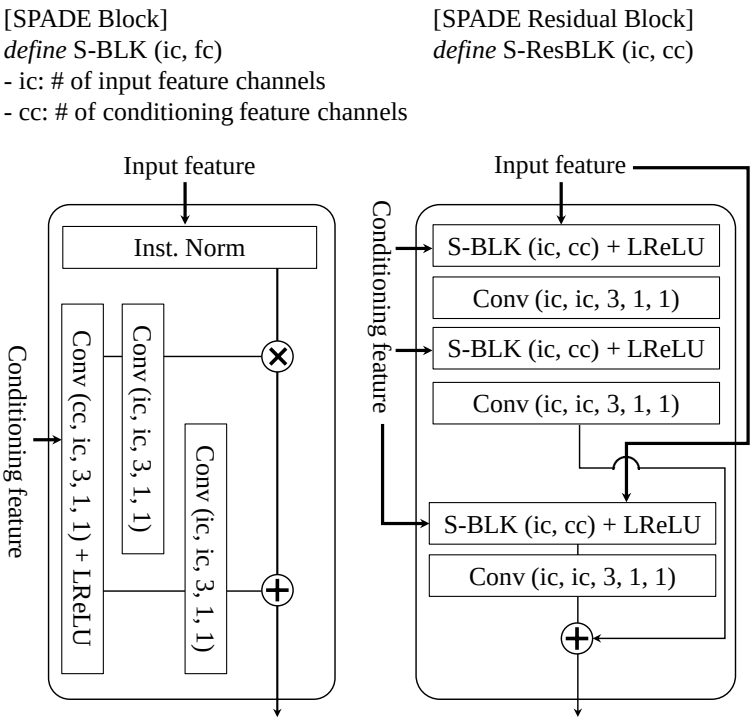


Figure 8.10. The description of SPADE and SPADE Residual blocks similar to [23]. Conv take as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.

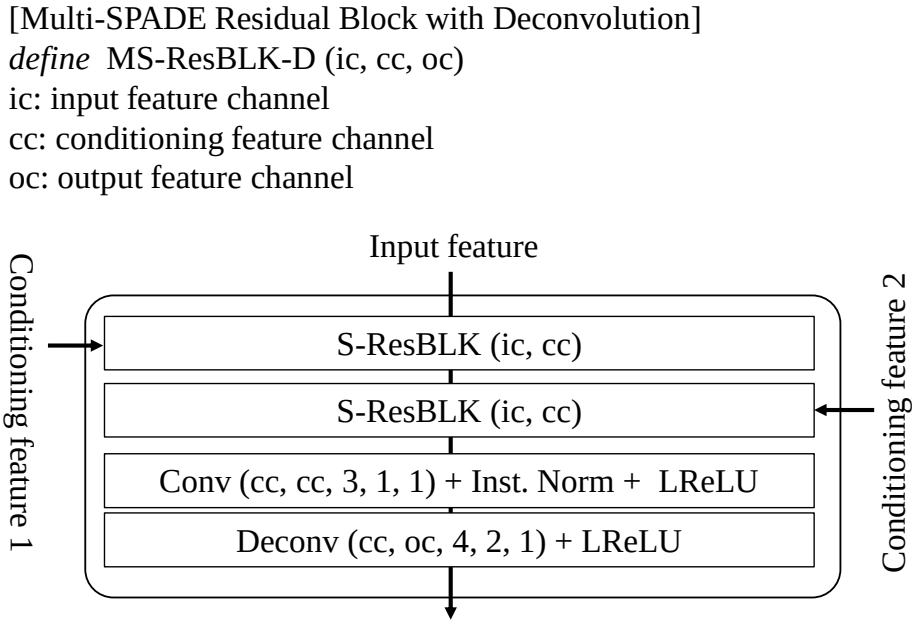


Figure 8.11. The description of Multi-Spade blocks similar to [24] where the details of S-ResBLK is described in Fig. 8.10. Conv and Deconv take as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.

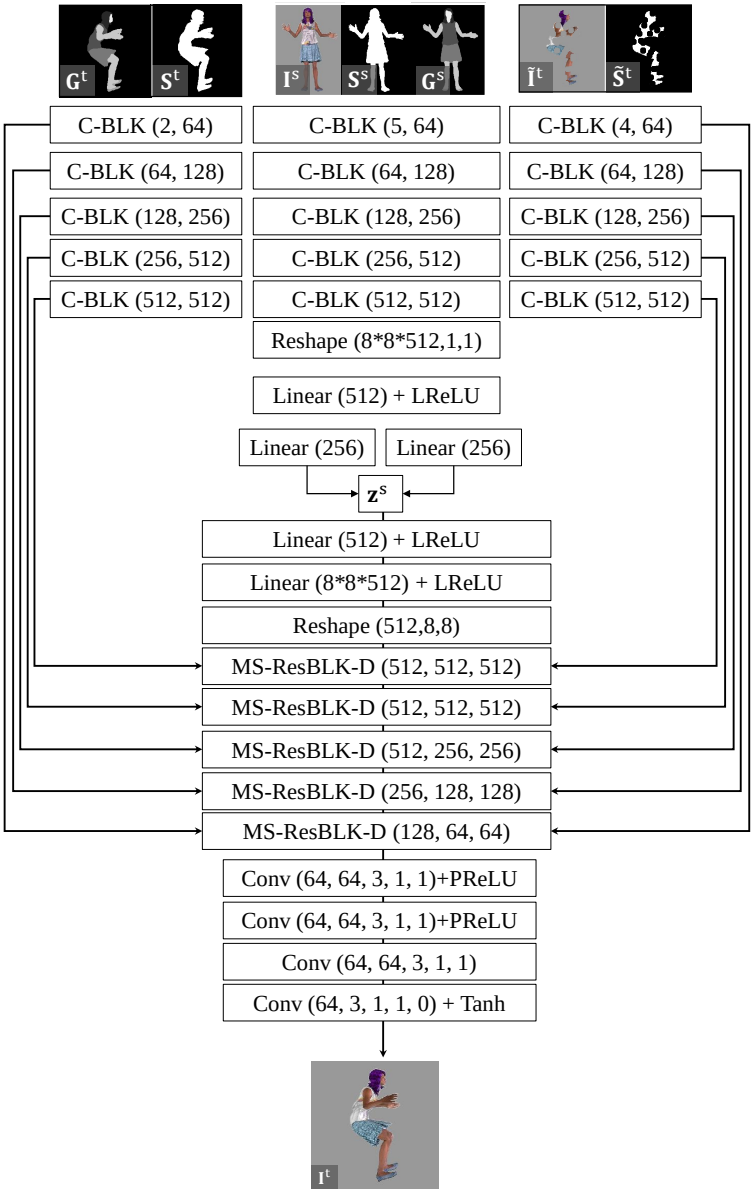


Figure 8.12. The details of our *RenderNet* where C-BLK and D-BLK are described in Fig 8.7, and MS-ResBLK-D is in Fig. 8.11. Conv takes as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.

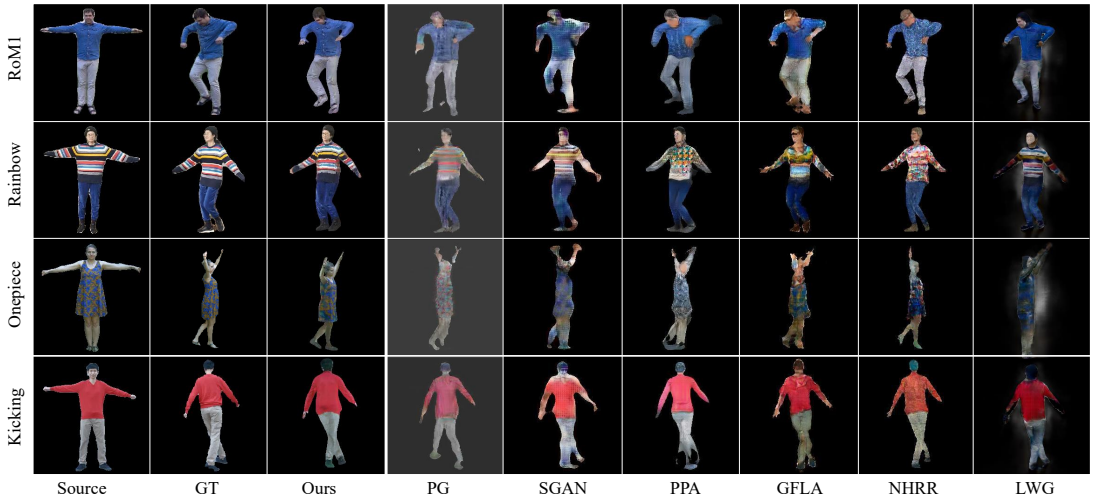


Figure 8.13. Qualitative comparisons of our approach with other baseline methods.

8.3 Experiments

In order to evaluate our approach, we collect eight sequences of the subjects in various clothing and motions from existing works [288, 67, 3, 289, 290] and capture two more sequences which include a person with more complex clothing style and motion than others. Each sequence contains 50 to 500 frames. We use one frame in the sequence as source image and estimated body poses from the rest of frames using a pose estimator [147] as a target pose sequence.

Baseline. We compare our method with related works including *PG* [179], *SGAN* [242], *PPA* [181], *GFLA* [196], *NHRR* [18], *LWG* [3]. Note that all these methods except LWG are not designed to handle background. We compare all the methods on foreground synthesis and conduct an additional comparison with LWG on the full image synthesis including both foreground and background. For a fair comparison, we train all the methods except LWG on 3D people dataset [2]. For training, LWG requires a SMPL model which is not provided by the 3D people dataset. Since registering a SMPL model to each 3D model in the 3D people dataset may introduce fitting error, we use the pretrained model provided by the authors, which are trained on the iPER dataset [3]. We also evaluate the methods with the pretrained models provided by the authors, which were trained on the Deep Fashion dataset [1]. In addition, we provide a qualitative comparison with *Photo Wake-Up* [283] which reconstructs a textured animatable 3D

	Maskman	Rainbow	RoM1	RoM2	Jumping
PG (DF)	2.01 / 4.24	2.14 / 4.41	2.22 / 4.48	1.81 / 4.31	2.33 / 4.32
SGAN (DF)	2.33 / 3.96	2.39 / 4.22	2.50 / 4.16	2.12 / 4.22	2.63 / 4.09
PPA (DF)	2.84 / 3.76	2.70 / 3.80	2.78 / 3.91	2.65 / 3.97	2.89 / 3.87
GFLA (DF)	1.96 / 3.86	1.64 / 3.93	2.19 / 3.89	1.50 / 3.99	2.01 / 3.85
NHHR (DF)	1.71 / 2.96	1.89 / 3.06	1.82 / 3.07	1.56 / 3.03	2.06 / 3.03
PG (3P)	3.45 / 4.46	3.28 / 4.58	3.38 / 4.37	3.30 / 4.57	3.87 / 4.68
SGAN (3P)	1.93 / 2.97	1.61 / 3.04	1.62 / 2.94	1.60 / 3.02	2.27 / 3.12
PPA (3P)	1.88 / 2.89	1.62 / 2.95	1.40 / 2.82	1.66 / 3.00	2.43 / 3.03
GFLA (3P)	1.90 / 2.92	1.59 / 3.05	1.53 / 2.91	1.71 / 2.95	2.11 / 3.06
NHHR (3P)	1.65 / 2.81	1.61 / 2.94	1.49 / 2.80	1.41 / 2.88	1.99 / 3.01
LWG (IPER)	2.66 / 3.54	2.16 / 3.57	2.17 / 3.49	2.24 / 3.73	4.11 / 3.49
Ours (3P)	**1.54** / **2.27**	**1.24** / **2.38**	**1.25** / **2.24**	**1.38** / **2.36**	**1.87** / **2.53**

Kicking	Onepiece	Checker	Rotation1	Rotation2	Average
2.15 / 4.49	2.43 4.66	2.07 / 4.25	1.74 / 4.18	2.58 / 4.47	2.15 / 4.38
2.49 / 4.29	2.67 / 4.25	2.34 / 3.99	1.89 / 3.93	2.74 / 4.22	2.43 / 4.13
2.88 / 3.94	3.21 / 4.05	2.26 / 3.76	2.26 / 3.75	3.01 / 3.77	2.74 / 3.86
2.05 / 3.96	2.23 / 3.94	1.74 / 3.84	1.60 / 3.88	1.92 / 3.89	1.88 / 3.90
1.68 / 3.11	2.16 / 3.16	1.48 / 2.94	1.80 / 3.02	2.77 / 3.11	1.89 / 3.05
2.98 / 4.36	3.26 / 4.63	3.03 / 4.45	3.67 / 4.41	2.95 / 4.06	2.93 / 4.45
1.56 / 2.98	1.82 / 3.06	1.53 / 3.03	1.56 / 3.01	1.65 / 2.84	1.71 / 3.00
1.33 / 2.88	1.86 / 2.95	1.49 / 2.88	1.38 / 2.84	1.40 / 2.69	1.64 / 2.89
1.42 / 2.97	1.60 / 3.03	1.64 / 2.97	1.64 / 2.93	1.65 / 2.80	1.68 / 2.96
1.00 / 2.85	1.78 / 2.98	1.65 / 2.94	1.39 / 2.88	1.67 / 2.75	1.56 / 2.88
2.42 / 3.57	2.31 / 3.65	2.40 / 3.57	2.14 / 3.64	2.20 / 3.48	2.48 / 3.57
1.08 / **2.19**	**1.23** / **2.32**	**1.09** / **2.24**	**1.00** / **2.19**	**1.12** / **2.16**	**1.28** / **2.29**

Table 8.1. Quantitative results with LPIPS (left, scale: $\times 10^1$) and CS (right) where the lower is the better. DF [1], 3P [2], and IPER [3] represent the name of dataset.

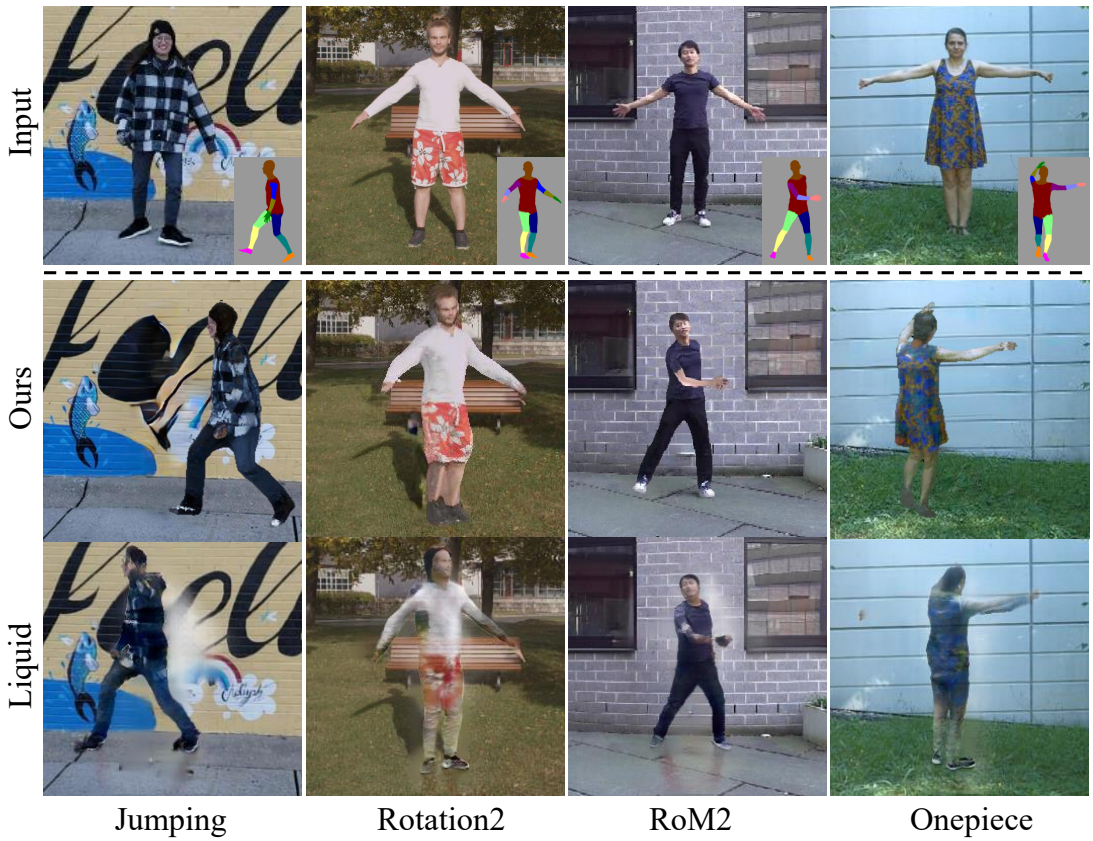


Figure 8.14. Qualitative comparison with LWG on the input images with background. The target pose is shown as inset.

model from a single image.

8.3.1 Comparisons

Qualitative Comparisons. We show the qualitative comparison with the baselines on the foreground synthesis in Fig. 8.13. Note that for the results in Fig. 8.13, all methods are trained on 3D people datasets. Our method significantly outperforms other baselines as preserving the facial identity, body shape and texture patterns of clothes over all the subjects with various challenging poses. Furthermore, compared to the baseline methods, our method generalizes better on the real data and achieves more realistic results that are close to the ground truth, although only synthetic data is used for training. We conduct a comparison with LWG on the full image synthesis, where

	Maskman	Rainbow	RoM1	RoM2	Jumping
R	1.64 / 2.31	1.48 / 2.43	1.41 / 2.30	1.53 / 2.44	2.00 / 2.54
GR	1.64 / 2.30	1.45 / 2.42	1.51 / 2.30	1.44 / 2.42	1.91 / 2.53
SR	1.57 / 2.26	1.30 / 2.42	1.31 / 2.24	1.41 / 2.37	1.89 / 2.54
SGR-$\mathbf{S}^{\mathrm{s}}$	1.58 / 2.29	1.33 / 2.41	1.26 / 2.26	1.43 / 2.39	1.99 / 2.54
SGR-$\mathbf{G}^{\mathrm{s}}$	1.66 / 2.30	1.38 / 2.39	1.31 / 2.32	1.48 / 2.35	1.89 / 2.51
SGR-$\bar{\mathbf{I}}^{\mathrm{t}}$	1.79 / 2.28	1.97 / 2.49	1.55 / 2.30	1.52 / 2.38	2.13 / 2.50
SGR-$\mathbf{z}^{\mathrm{s}}$	1.57 / 2.27	1.31 / 2.40	1.25 / 2.26	1.42 / 2.38	1.90 / 2.52
SGR-L_{KL}	1.54 / 2.27	1.25 / 2.38	1.27 / 2.25	1.40 / 2.38	1.88 / 2.55
SGR-$\mathbf{A}$	1.59 / 2.28	1.28 / 2.40	1.31 / 2.26	1.40 / 2.38	1.86 / 2.51
SGR (**full**)	**1.54** / **2.27**	**1.24** / **2.38**	**1.25** / **2.24**	**1.38** / **2.36**	**1.87** / **2.53**
SGR+2view	1.50 / 2.25	1.22 / 2.38	1.21 / 2.23	1.33 / 2.36	1.80 / 2.51
SGR+4view	1.49 / 2.25	1.21 / 2.38	1.21 / 2.23	1.33 / 2.35	1.80 / 2.51

Kicking	Onepiece	Checker	Rotation1	Rotation2	Average
1.16 / 2.18	1.36 / 2.34	1.41 / 2.31	1.22 / 2.31	1.62 / 2.33	1.48 / 2.35
1.40 / 2.24	1.24 / 2.35	1.39 / 2.29	1.21 / 2.30	1.60 / 2.32	1.47 / 2.35
1.17 / 2.20	1.24 / 2.33	1.11 / 2.24	1.05 / 2.23	1.25 / 2.22	1.33 / 2.31
1.18 / 2.23	1.29 / 2.35	1.10 / 2.36	1.05 / 2.23	1.24 / 2.20	1.35 / 2.32
1.18 / 2.23	1.31 / 2.40	1.31 / 2.31	1.19 / 2.28	1.42 / 2.30	1.41 / 2.34
1.31 / 2.23	1.79 / 2.39	1.49 / 2.31	1.15 / 2.22	1.50 / 2.21	1.62 / 2.33
1.15 / 2.19	1.29 / 2.31	1.11 / 2.22	1.05 / 2.19	1.24 / 2.23	1.32 / 2.30
1.13 / 2.19	1.25 / 2.32	1.09 / 2.24	1.04 / 2.20	1.15 / 2.19	1.30 / 2.30
1.23 / 2.21	1.32 / 2.33	1.14 / 2.25	1.15 / 2.23	1.28 / 2.20	1.36 / 2.31
1.08 / **2.19**	**1.23** / **2.32**	**1.09** / **2.24**	**1.00** / **2.19**	**1.12** / **2.16**	**1.28** / **2.29**
1.15 / 2.17	1.20 / 2.31	1.07 / 2.23	0.97 / 2.16	1.06 / 2.14	1.25 / 2.28
1.12 / 2.17	1.20 / 2.31	1.07 / 2.23	0.98 / 2.16	1.07 / 2.14	1.24 / 2.27

Table 8.2. Quantitative results of our ablation study. We denote our complete model with a single image as input as SGR(**full**).

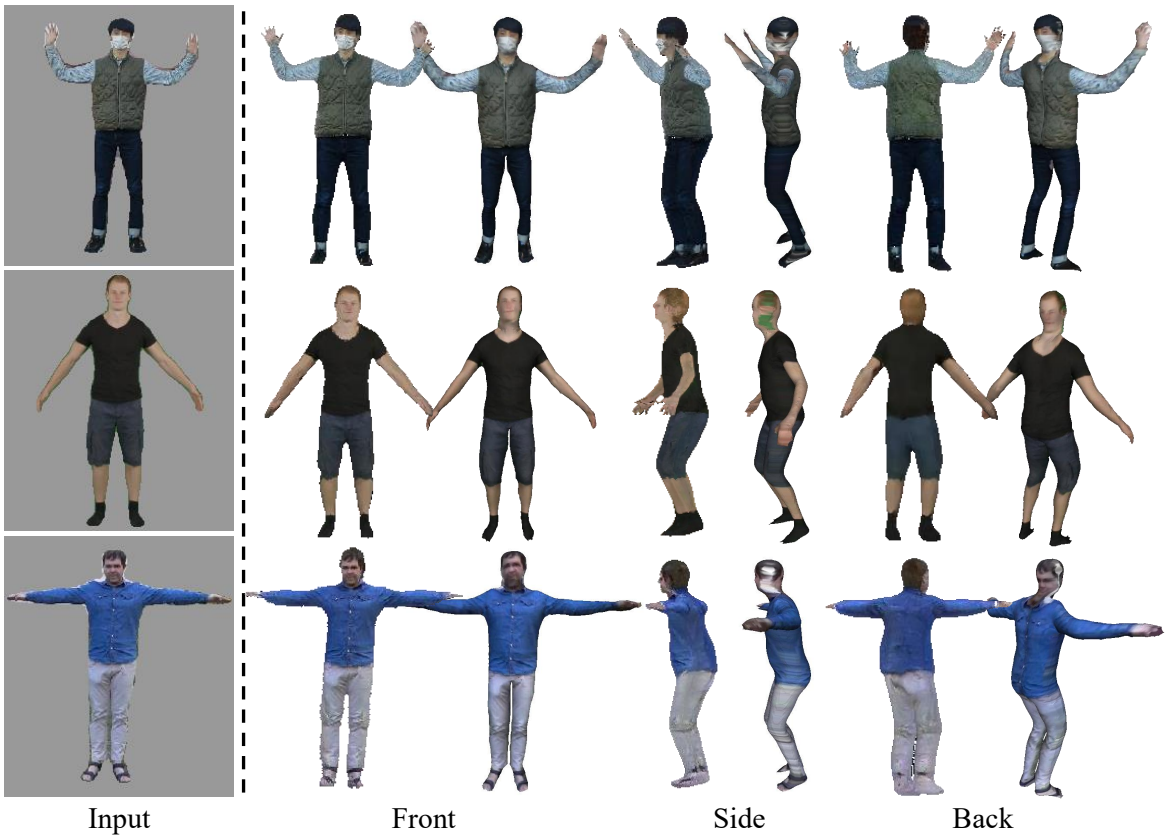


Figure 8.15. Qualitative comparison of ours (left) with Photo Wake-Up (right).

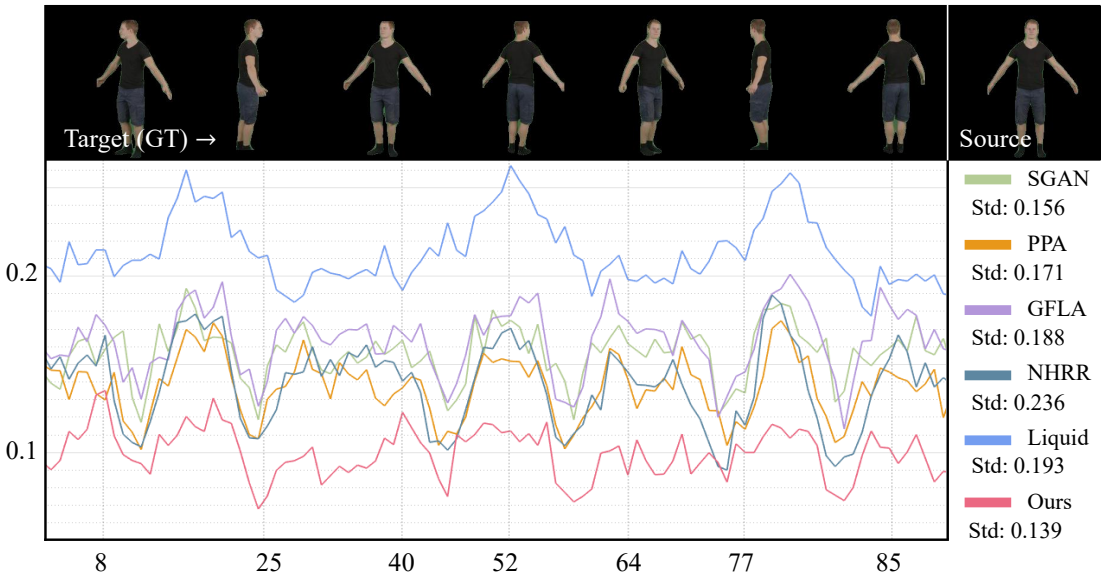


Figure 8.16. The accuracy graph for the entire frames of a video. x-axis and y-axis represent time instance and LPIPS, respectively.

our method can synthesize higher quality foreground as well as background, as shown in Fig. 8.14. Compared to Photo Wake-Up in Fig. 8.15, we can render the better textures on the right and back side of the person.

Quantitative Comparisons. We measure the quality on testing results with two metrics: LPIPS [63] and CS [277] where both metrics measure the similarity of the generated image with ground truth based on the deep features, and CS can handle the non-aligned two images. As shown in Table 8.1, our method outperforms all baseline methods over almost all the sequences in LPIPS and CS. In Kicking, our method performs the second best in LPIPS metric mainly due to the misalignment with the ground truth originated from the pose estimation error. In Fig. 8.16, we measure temporal stability of the synthesized animations with the standard deviation of the LPIPS scores with respect to all the frames, where our results show the best temporal stability.

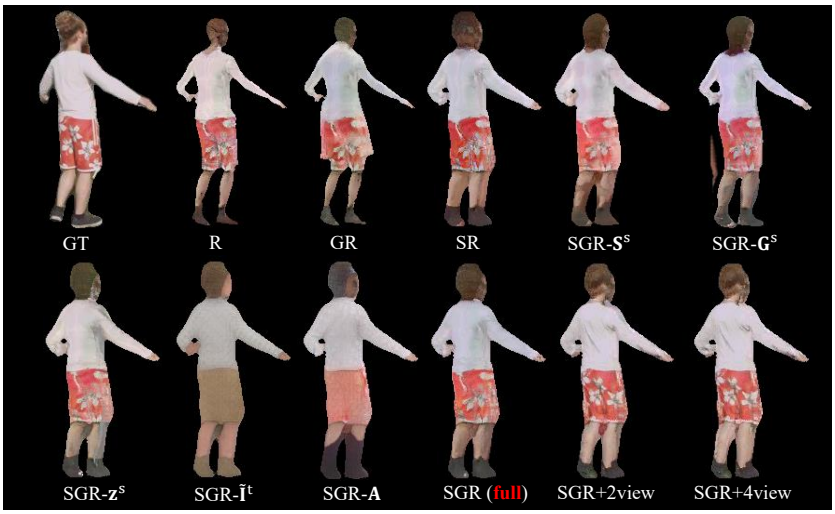


Figure 8.17. Qualitative results of our ablation study.

8.3.2 Ablation Study

We study the importance of each module in our pose transfer pipeline where we term "S", "G", and "R" as *SilNet*, *GarNet*, and *RenderNet*, and our full model as *SGR*.

1) We analyze the effectiveness of our modular network by removing each from SGR where the intermediate results are also removed from the entire pipeline: *R*, *SR*, and *GR*.

2) We evaluate the impact of using silhouette mask and garment label from the source by removing each of them from the entire pipeline: *SGR*-$\mathbf{S}^{\mathrm{s}}$ and *SGR*-$\mathbf{M}^{\mathrm{s}}$.

3) We investigate the improvement factor on the *RenderNet*: *SGR*-$\mathbf{z}^{\mathrm{s}}$, *SGR*-$\widetilde{\mathbf{I}}^{\mathrm{t}}$, and *SGR*-$L_{\mathrm{KL}}$. For *SGR*-$L_{\mathrm{KL}}$, we represent the latent space with fully connected layers. On top of that, we investigate the impact of reconstructing a complete UV map: *SGR*-**A**. In this case, we create the pseudo target image $\widetilde{\mathbf{I}}^{\mathrm{t}}$ by directly warping the source image to the target.

4) Finally, we show that our method is readily extendable to the multiview setting by unifying all the pixels from multiple images in the coherent UV maps. For this, we choose two or four frames from the testing videos that include salient body sides, *e.g.,* front, back, right, and left: *SGR*+2view and *SGR*+4view.

We summarize the results of our ablation study in Table 8.2 and the qualitative results

are shown in Fig. 8.17. Separating the silhouette prediction module from rendering network brings out notable improvement, and the predicted garment labels $\mathbf{G}^{\mathrm{t}}$ further improve the results, *e.g.,* clear boundary between different classes. Without the garment labels from the source $\mathbf{G}^{\mathrm{s}}$ the performance is largely degraded due to the misclassified body parts. Conditioning the style code $\mathbf{z}^{\mathrm{s}}$ from the source improves the generation quality, *e.g.,* seamless inpainting. Conditioning the pseudo images $\widetilde{\mathbf{I}}^{\mathrm{t}}$ warped from the coherent UV maps $\mathbf{A}$ plays the key role to preserve the subject's appearance in the generated image. Leveraging multiview images better can preserve the clothing texture, *e.g.,* the flower patterns in the subject's half pants.

8.3.3 User Study

We evaluate the qualitative impact of our method by a user study with 25 videos where each video shows a source image and animated results. Four videos compare our method to LWG on the scenes with a background. 21 videos are without background (15 of them compare our method to randomly-chosen four baselines, excluding ground truth, and 6 videos include ground truth). In our user study, three questions are asked: Q1: Which video looks most realistic including temporal coherence? Q2: Which video preserves the identity best including facial details, shape, and overall appearance? Q3: In which video, the background is preserved better across the frames (only for the case of scenes with background)? For each method, we measure the performance based on the number of entire votes divided by the number of participants and the number of occurrence in the questionnaires. The full results are shown in Fig. 8.18. 47 people participated in total. The first question was answered in 84.3% and 93.0% of the cases in favour of our method with and without the ground truth sequence, respectively, and the second question 84.1% and 94.2%. Moreover, these numbers strongly correlate with the identity-preserving properties of our method. In the third question, the background is preserved better in our method than LWG in 96.8 % of the answers. The results show that our method outperforms other state of the art, and our animations are in many cases qualitatively comparable to real videos of the subjects. The choice between a real video and our animation did not fall easy because the ground-truth video often contains noisy boundary originated from the person segmentation error while the generated person images from our method shows the clear boundary. Finally, our technique preserves

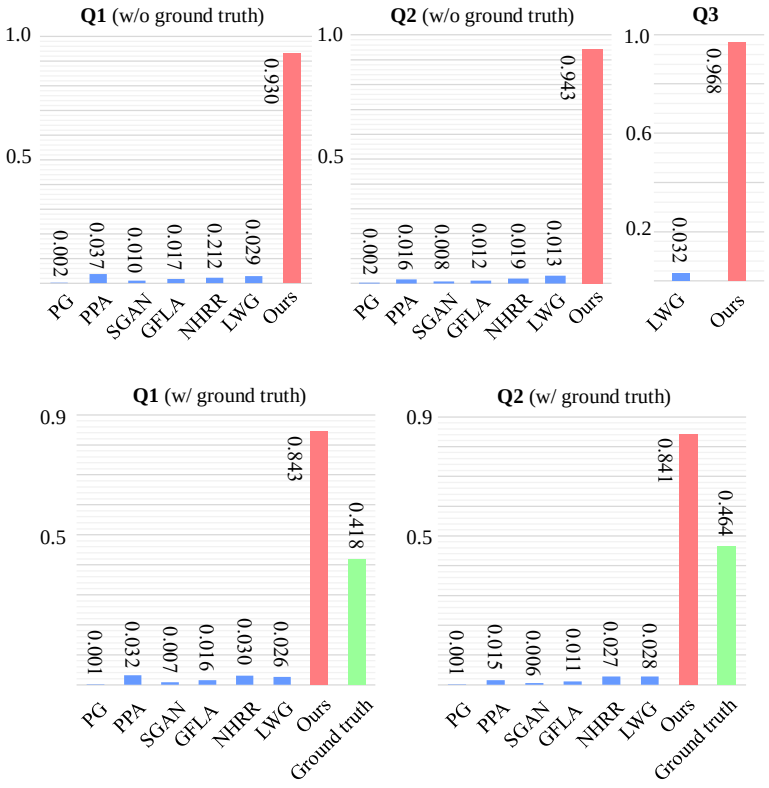


Figure 8.18. The full results of the user study where x-axis represents the number of votes for the associated method which is normalized by the number of participants and the number of occurrence in the questionnaires. Q1, Q2, and Q3 represent the question type. Our results were often ranked as more realistic than the real videos because they involve a significant boundary noise from the person segmentation error while our method produces the human animation with clean boundary.

the background better compared to LGW, in the opinion of respondents (96.8% of the answers). The user study shows that our method significantly outperforms the state of the arts in terms of synthesis quality, temporal consistency and generalizability. Also, our results were often ranked as more realistic than the ground truth videos.

8.3.4 Limitations

Our method has several limitations. Although the unified representation of appearance and its labels allow us to synthesize temporally consistent results, it prevents from generating realistic physical effects such as pose-dependent clothing secondary motion, wrinkles, shading, and view-dependent lighting. Because of non-end-to-end nature of our method, the errors from the pre-processing step, *e.g.,* person and garment segmentation, and pose estimation, cannot be corrected by our pose transfer network.

8.4 Summary

We introduce a new pose transfer framework to animate humans from a single image. We addressed the core domain gap challenge for the testing data in the wild by designing a new compositional pose transfer network that predicts silhouette, garment labels, and textures in series, which are learned from synthetic data. In inference time, we reconstruct coherent UV maps by unifying the source and synthesized images, and utilize these UV maps to guide the network to create coherent human animation. The evaluation on diverse subjects demonstrates that our framework works well on the unseen data without any fine-tuning and preserves the identity and texture of the subject as well as background in a temporally coherent way, showing a significant improvement over the state-of-the-arts.

Chapter 9

Learning Motion-Dependent Appearance for High-Fidelity Rendering of Dynamic Humans from a Single Camera

We express ourselves by moving our body that drives a sequence of natural secondary motion, e.g., dynamic movement of dress induced by dancing as shown in Figure 9.1. This secondary motion is the resultant of complex physical interactions with the body, which is, in general, *time-varying*. This presents a major challenge for plausible rendering of dynamic dressed humans in applications such as video based retargetting or social presence. Many existing approaches such as pose-guided person image generation [26] focus on static poses as a conditional variable. Despite its promising rendering quality, it fails to generate a physically plausible secondary motion, e.g., generating the same appearance for fast and slow motions.

One can learn the dynamics of the secondary motion from videos. This, however, requires a tremendous amount of data, i.e., videos depicting all possible poses and associated motions. In practice, only a short video clip is available, e.g., the maximum length of videos in social media (e.g., TikTok) are limited to 15-60 seconds. The learned representation is, therefore, prone to overfitting.

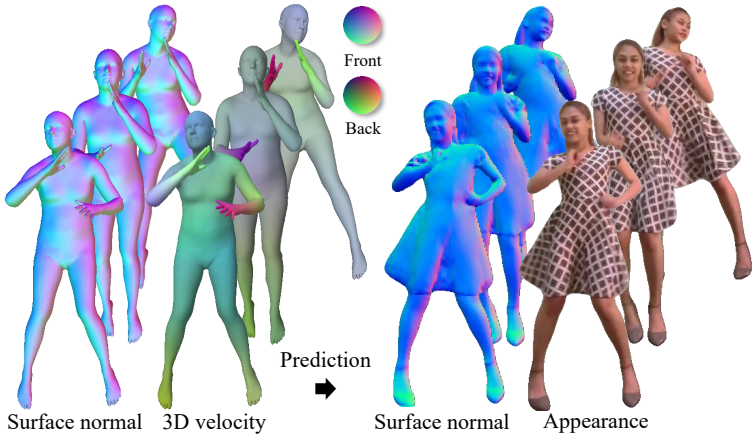


Figure 9.1. Given surface normal and velocity of a 3D body model, our method synthesizes subject-specific surface normal and appearance. We specifically focus on synthesis of plausible dynamic appearance by learning an effective 3D motion descriptor.

In this paper, we address the fundamental question of "can we learn a representation for dynamics given a limited amount of observations?". We argue that a meaningful representation can be learned by enforcing an equivariant property—a representation is expected to be transformed in the way that the body pose is transformed. With the equivariance, we model the dynamics of the secondary motion as a function of spatial and time derivative of the 3D body. We construct this representation by re-arranging 3D features in the canonical coordinate system of the body surface, i.e., the UV map, which is invariant to the choice of the 3D coordinate system.

The UV map also captures the semantic meaning of body parts since each body part is represented by a UV patch. The resulting representation is compact and discriminative compared to the 2D pose representations that often suffer from geometric ambiguity due to 2D projection.

We observe that two dominant factors significantly impact the physicality of the generated appearance. First, the silhouette of dressed humans is transformed according to the body movement and the physical properties (e.g., material) of the individual garment types (e.g., top and bottom garments might undergo different deformations). Second, the local geometry of the body and clothes is highly correlated, e.g., surface normals of T-shirt and body surface, which causes appearance and disappearance of

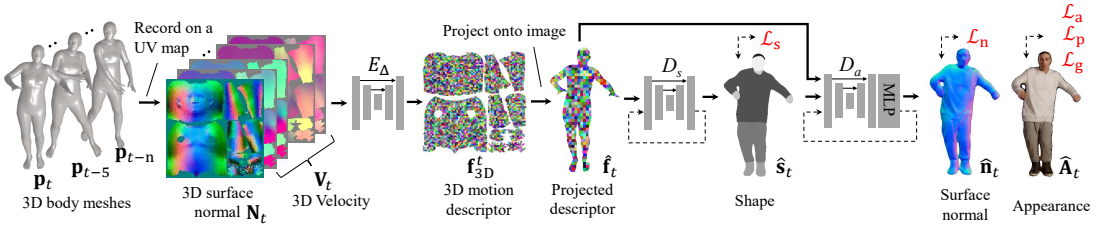


Figure 9.2. The overview of our human rendering pipeline. Given a set of time-varying 3D body meshes $\{\mathbf{P}_t, .., \mathbf{P}_{t-n}\}$ obtained from a monocular input video, we aim to synthesize high-fidelity appearance of a dressed human. We learn an effective 3D body pose and motion representation by recording the surface normal $\mathbf{N}^t$ of the posed 3D mesh at time t and the body surface velocity $\mathbf{V}^t$ over several past times in the spatially aligned UV space. We define an encoder E_Δ which is designed to reconstruct 3D motion descriptors $\mathbf{f}_{3D}^t$ that encode the spatial and temporal relation of the 3D body meshes. Given a target 3D body configuration, we project $\mathbf{f}_{3D}^t$ onto the image space which are then utilized by our compositional networks (D_s and D_a) to predict a shape with semantic labels, surface normal, and final appearance.

folds and wrinkles. To incorporate these factors, we propose a compositional decoder that breaks down the final appearance rendering into modular subtasks. This decoder predicts the time-varying semantic maps and surface normals as intermediate representations. While the semantic maps capture the time-varying silhouette deformations, the surface normals are effective in synthesizing high quality textures, which further enables re-lighting. We combine these intermediate representations to produce the final appearance.

Our experiments show that our method can generate a temporally coherent video of an unseen secondary motion from novel views given a single view training video. We conduct thorough comparisons with various state-of-the-art baseline approaches. Thanks to the discriminative power, our representation demonstrates superior generalization ability, consistently outperforming previous methods when trained on shorter training videos. Furthermore, our method shows better performance in handling complex motion sequences including 3D rotations as well as rendering consistent views in applications such as free-viewpoint rendering. The intermediate representations predicted by our method such as surface normals also enable applications such as relighting which are otherwise not applicable.

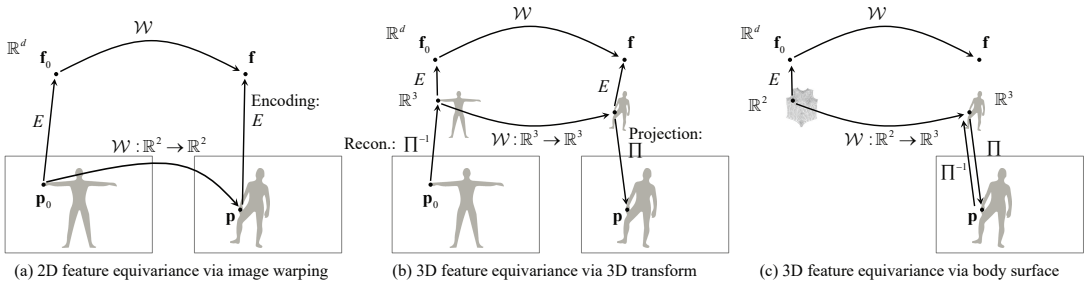


Figure 9.3. We apply equivariance to learn a compact representation. (a) In 2D, the feature $\mathbf{f} = E(\mathbf{p})$ is expected to be transformed to the feature of the neutral pose, $\mathbf{f}_0 = E(\mathcal{W}^{-1}\mathbf{p})$ by a coordinate transform $\mathcal{W}$, e.g., image warping. This eliminates the necessity of learning the encoder E, i.e., the appearance of the pose $\mathbf{p}$ is generated by warping the appearance of the neutral pose. (b) Equivariance in 3D can be applied by incorporating 3D body reconstruction Π^{-1} where the feature is expected to be transformed by the 3D warping $\mathcal{W}$, e.g., skinning. (c) We use the canonical body surface coordinate (UV coordinate) to represent the feature coordinate transformation.

9.1 Method

Given a monocular video of a person in motion and the corresponding 3D body fit estimates, we learn a motion representation to describe the time-varying appearance of the secondary motion induced by body movement (Section 9.1.1). We propose a multitask compositional renderer (Section 9.1.2) that uses this representation to render the subject-specific final appearance of moving dressed humans. Our renderer first predicts two intermediate representations including time-varying semantic maps that capture garment specific silhouette deformations and surface normals that capture the local geometric changes such as folds and wrinkles. These intermediate representations are combined to synthesize the final appearance. We obtain the 3D body fits estimates from the input video using a new model-based tracking optimization (Supplementary material). The overview of our rendering framework is shown in Figure 9.2.

9.1.1 Equivariant 3D Motion Descriptor

We cast the problem of human rendering as learning a representation via a feature encoder-decoder framework:

$$\mathbf{f} = E(\mathbf{p}), \quad \mathbf{A} = D(\mathbf{f}), \tag{9.1}$$

where an encoder E takes as an input a representation of a posed body, $\mathbf{p}$ (e.g., 2D sparse or dense keypoints or 3D body surface vertices), and outputs per-pixel features $\mathbf{f}$ that can be used by the decoder D to reconstruct the appearance $\mathbf{A} \in [0,1]^{w \times h \times 3}$ of the corresponding pose where w and h are the width and height of the output image (appearance). We first discuss how E can be modeled to render static appearance, then introduce our 3D motion descriptor to render time-varying appearance with secondary motion effects.

Learning a representation from Equation (9.1) from a limited amount of data is challenging because both encoder and decoder need to memorize every appearance in relation to the corresponding pose, $\mathbf{A} \leftrightarrow \mathbf{p}$. To address the data challenge, one can use an *equivariant* geometric transformation, $\mathcal{W}$, such that a feature is expected to be transformed in the way that the body pose is transformed:

$$E(\mathcal{W}\mathbf{x}) = \mathcal{W}E(\mathbf{x}). \tag{9.2}$$

where $\mathbf{x}$ is an arbitrary pose. A naive encoder that satisfies this equiavariance learns a constant feature $\mathbf{f}_0$ by warping any $\mathbf{p}$ to a neutral pose $\mathbf{p}_0$:

$$\mathbf{f}_0 = E(\mathcal{W}^{-1}\mathbf{p}) = \text{const.}, \quad \mathbf{A} = D(\mathcal{W}\mathbf{f}_0), \tag{9.3}$$

where $\mathbf{p} = \mathcal{W}\mathbf{p}_0$. Figure 9.3(a) and (b) illustrate cases where $\mathcal{W}$ is defined as image warping in 2D or skinning in 3D respectively. $\mathbf{f}$ can be derived by warping $\mathbf{p}$ to the T-pose, $\mathcal{W}^{-1}\mathbf{p}$ of which feature can be warped back to the posed feature before decoding, $D(\mathcal{W}E(\mathcal{W}^{-1}\mathbf{p}))$. Since $\mathbf{f}_0$ is constant, the encoder E does not need to be learned. One can only learn the decoder D to render a static appearance.

To model the time-varying appearance for the secondary motion that depends on both body pose and motion, one can extend Equation (9.3) to encode the spatial and

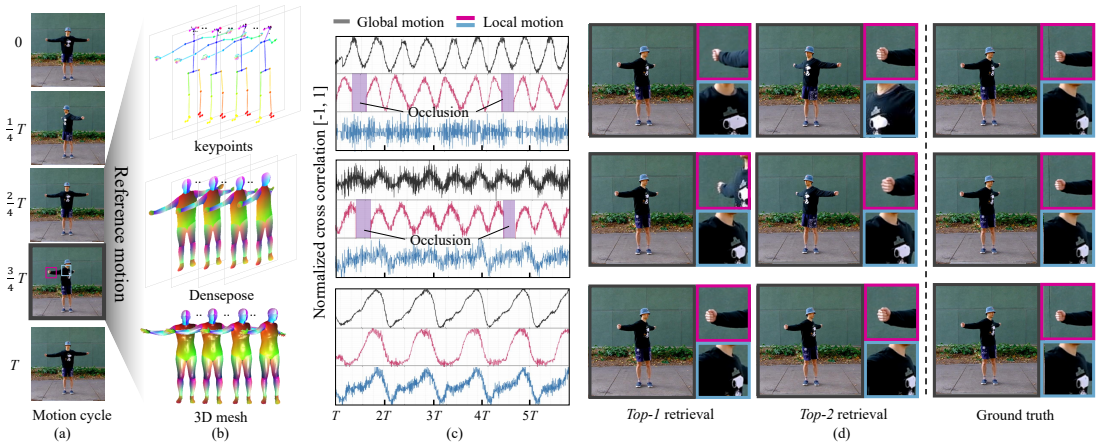


Figure 9.4. We show the strength of our 3D motion descriptor using a toy example. Given a video of a person rotating his body from left to right multiple times, we associate the first cycle of the motion (*i.e.*, $0 \sim T$) to the remaining cycles ($T \sim 6T$). As a proof-of-concept, we use a nearest neighbor classifier to model D. (b) We represent the motion descriptor using (top) 2D keypoints [25], (middle) 2D dense UV coordinates [13], (bottom) and 3D body mesh [17]. (c) We measure the similarity in motion descriptor for entire body (gray), local hand (pink) and upper torso (blue) using normalized cross correlation (NCC) where multiple peaks within a cycle indicate ambiguity of the descriptor. (d) Given the motion descriptors, we retrieve relevant image patches. While the 3D motion descriptors identify the image patches similar to the ground truth, due to the depth ambiguity, the 2D motion descriptors result in ambiguous matches. Furthermore, the 2D motion descriptors are not well defined in case of occlusions.

temporal gradients as a residual feature encoding:

$$\begin{aligned} \mathbf{f} &= E\left(\mathbf{p}, \frac{\partial \mathbf{p}}{\partial x}, \frac{\partial \mathbf{p}}{\partial t}\right) \approx E(\mathbf{p}) + E_{\Delta}\left(\frac{\partial \mathbf{p}}{\partial x}, \frac{\partial \mathbf{p}}{\partial t}\right) \\ \iff \mathbf{f}_0 &= E(\mathcal{W}^{-1}\mathbf{p})\text{const.} + E_{\Delta}\left(\mathcal{W}^{-1}\frac{\partial \mathbf{p}}{\partial x}, \mathcal{W}^{-1}\frac{\partial \mathbf{p}}{\partial t}\right), \end{aligned} \tag{9.4}$$

where $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial t}$ are the spatial and temporal derivatives of the posed body, respectively. The spatial derivatives essentially represent the pose corrective deformations [291, 17]. The temporal derivatives denote the body surface velocity which results in secondary motion. Since these spatial and temporal gradients are no longer constant, one needs to learn an encoder E_{Δ} to encode the residual features.

In this paper, we use a 3D representation of the posed body lifted from an image by leveraging recent success in single view pose reconstruction [153, 292, 293]. Hence,

spatial and temporal derivatives of the body pose correspond to the surface normals and body surface velocities, respectively:

$$\mathbf{f}_{3D} = E_{\Delta}(\mathcal{W}^{-1}\mathbf{N}, \mathcal{W}^{-1}\mathbf{V}), \quad \mathbf{A} = D(\Pi\mathcal{W}\mathbf{f}_{3D}), \tag{9.5}$$

where $\mathbf{N} = \frac{\partial \mathbf{p}}{\partial X} \in R^{m\times 3}$ is the 3D surface normal, and $\mathbf{V} = \frac{\partial \mathbf{p}}{\partial t} \in R^{m\times 3}$ represents the instantaneous velocities of the m vertices in the body surface. We model the geometric transformation function $\mathcal{W}$ to warp an arbitrary 3D pose $\mathbf{p}$ to a canonical representation, $\mathbf{p}_0$. We record $\mathbf{f}_{3D}$ in a spatially aligned 2D positional map, specifically the UV map of the 3D body mesh where each pixel contains the 3D information of a unique point on the body mesh surface. This enables us to leverage 2D CNNs to apply local convolution operations to capture the relationship between neighboring body parts [**?**]. Therefore, $\mathbf{f}_{3D} \in R^{m\times d}$ called *3D motion descriptor* is the feature defined in the UV coordinates where d is the dimension of the per-vertex 3D feature. $\mathbf{f} = \Pi\mathcal{W}\mathbf{f}_{3D}$ is the projected 3D feature in the image coordinates where Π is a coordinate transformation that transports the features defined in the UV space to the image plane via the dense UV coordinates of the body mesh.

The key advantage of the 3D motion descriptor over commonly used 2D sparse [28] or dense [4] keypoint representations is discriminativity. Consider a toy example of a person rotating his body left to right multiple times. Given one cycle (i.e., 0-T) of such a motion as input (Figure 9.4(a)), assume we want to synthesize the appearance of the person performing the repetitions of the same motion (i.e., cycles T-5T). As a proof of concept, we model D using a nearest neighbor classifier to retrieve the relevant image patches (top two patches) for each body part from the reference motion based on the correlation of the motion descriptors as shown in Figure 9.4(c). Due to the inherent depth ambiguity, multiple 3D motion trajectories yield the same 2D projected trajectory [294]. Hence, the 2D motion descriptors using sparse (Figure 9.4(b), top) and dense (Figure 9.4(b), middle) 2D keypoints confuse the directions of out-of-plane body rotation, resulting in erroneous nearest neighbor retrievals as shown in Figure 9.4(d). Furthermore, 2D representations entangle the viewpoint and pose into a common feature. This not only avoids a compact representation (e.g., the same body motion maps to different 2D trajectories with respect to different viewpoints and yields different features) but also suffers from occlusions in the image space. In our example in Figure 9.4(c), top and

bottom, the upper arm is occluded in portions of the input video denoted with the purple blocks, hence no reliable local motion descriptors can be computed at those time instances. In contrast, our 3D motion descriptor is highly discriminative, which does not confuse the direction of body rotations, resulting in accurate image patch retrievals.

9.1.2 Multitask Compositional Decoder

Given 3D motion features, the decoder D still needs to learn to generate diverse and plausible secondary motion, which is prone to overfitting given a limited amount of data. We integrate the following properties that can mitigate this challenge. (1) Composition: we design the decoder using a composition of modular functions where each modular function is learned to generate physically and semantically meaningful intermediate representations. Learning each modular function is more accurate than learning an end-to-end decoder as a whole as shown in our ablation study (Section 9.3.3); (2) Multi-task: each intermediate representation receives its own supervision signals resulting in multi-task learning. The motion features, $\mathbf{f}_{\text{3D}}$, are shared by all intermediate modules resulting in a compact representation; (3) Recurrence: each module is modeled as an autoregressive network, which allows learning the dynamics rather than memorizing the pose-specific appearance.

Our decoder is a composition of two modular functions:

$$D = D_a \circ D_s, \tag{9.6}$$

where D_s and D_a are the functions that generate the shape with semantic maps and the appearance.

D_s learns the dynamics of the 2D shape:

$$\widehat{\mathbf{s}}_t = D_s(\widehat{\mathbf{s}}_{t-1}; \widehat{\mathbf{f}}_t) \tag{9.7}$$

where $\widehat{\mathbf{f}}_t = \Pi \mathcal{W}_t \mathbf{f}_{\text{3D}}^t$ is the projected features onto the image at time t, and $\widehat{\mathbf{s}}_t \in \{0, \cdots, L\}^{w \times h}$ is the predicted shape with semantics where L is the number of semantic categories. In our experiments we set $L = 7$ (background, top clothing, bottom clothing, face, hair, skin, shoes).

D_a learns the dynamics of appearance given the shape and 3D motion descriptor:

$$\widehat{\mathbf{A}}_t, \widehat{\mathbf{n}}_t = D_a(\widehat{\mathbf{A}}_{t-1}, \widehat{\mathbf{n}}_{t-1}; \widehat{\mathbf{s}}_t, \widehat{\mathbf{f}}_t), \tag{9.8}$$

where $\widehat{\mathbf{A}}_t \in R^{w \times h \times 3}$ and $\widehat{\mathbf{n}}_t \in R^{w \times h \times 3}$ are the generated appearance and surface normals at time t.

We learn the 3D motion descriptor as well as the modular decoder functions by minimizing the following loss:

$$\mathcal{L} = \sum_{\mathbf{P},\mathbf{A} \in \mathcal{D}} \mathcal{L}_{\mathrm{a}} + \lambda_{\mathrm{s}}\mathcal{L}_{\mathrm{s}} + \lambda_{\mathrm{n}}\mathcal{L}_{\mathrm{n}} + \lambda_{\mathrm{p}}\mathcal{L}_{\mathrm{p}} + \lambda_{\mathrm{g}}\mathcal{L}_{\mathrm{g}}, \tag{9.9}$$

where $\mathcal{L}_{\mathrm{a}}$, $\mathcal{L}_{\mathrm{s}}$, $\mathcal{L}_{\mathrm{n}}$, $\mathcal{L}_{\mathrm{p}}$, $\mathcal{L}_{\mathrm{g}}$ are the appearance, shape, surface normal, perceptual similarity, and generative adversarial losses, and λ_{s}, λ_{n}, λ_{p}, and λ_{g} are their weights, respectively. We set $\lambda_{\mathrm{s}} = 10$, $\lambda_{\mathrm{n}} = \lambda_{\mathrm{p}} = 1$ and $\lambda_{\mathrm{g}} = 0.01$ in our experiments. $\mathcal{D}$ is the training dataset composed of the ground truth 3D pose $\mathbf{P}$ and its appearance $\mathbf{A}$.

$$\begin{aligned}
\mathcal{L}_{\mathrm{a}}(\mathbf{P}, \mathbf{A}) &= \|\widehat{\mathbf{A}} - \mathbf{A}\|, \\
\mathcal{L}_{\mathrm{s}}(\mathbf{P}, \mathbf{A}) &= \|\widehat{\mathbf{s}} - S(\mathbf{A})\|, \\
\mathcal{L}_{\mathrm{n}}(\mathbf{P}, \mathbf{A}) &= \|\widehat{\mathbf{n}} - N(\mathbf{A})\|, \\
\mathcal{L}_{\mathrm{p}}(\mathbf{P}, \mathbf{A}) &= \sum_i \|VGG_i(\widehat{\mathbf{A}}) - VGG_i(\mathbf{A})\|, \\
\mathcal{L}_{\mathrm{g}}(\mathbf{P}, \mathbf{A}) &= E_{S(\mathbf{A}),\mathbf{A}}[\log(D^{\star}(S(\mathbf{A}), \mathbf{A})] + \\
&\qquad E_{S(\mathbf{A}),\widehat{\mathbf{A}}}[\log(1 - D^{*}(S(\mathbf{A}), \widehat{\mathbf{A}})],
\end{aligned}$$

where $\widehat{\mathbf{A}}$, $\widehat{\mathbf{s}}$, and $\widehat{\mathbf{n}}$ are the generated appearance, shape, and surface normal, respectively. S and N are the shape [56] and surface normal estimates [295], and VGG is the feature extractor that computes perceptual features from conv-i-2 layers in VGG-16 networks [296], D^{*} is the PatchGAN discriminator [279] that validates the plausibility of the synthesized image conditioned on the shape mask.

9.1.3 Model-based Monocular 3D Pose Tracking

While there has been significant improvements in monocular 3D body estimation [293, 147, 292], we observe that predicting accurate and temporally coherent 3D body sequences is still challenging, which inhibit to reconstruct high-quality 3D motion descriptors. Hence, we devise a new optimization framework that learns a tracking function to address this challenge.

Given a video of a moving person, we represent $\mathbf{p}$ as the posed 3D body at each frame. Specifically, we predict the parameters of the template SMPL model [17], i.e., $\mathbf{p} = SMPL(\boldsymbol{\theta}, \boldsymbol{\beta})$, where $SMPL$ is a function that takes the pose $\boldsymbol{\theta} \in R^{72}$ and shape $\boldsymbol{\beta} \in R^{10}$ parameters and provides the vertex locations of the 3D posed body.

$$\boldsymbol{\theta}_t, \mathbf{C}_t = f_{\text{track}}(\mathbf{A}_t), \tag{9.10}$$

where f_{track} is the tracking function, $\mathbf{A}_t$ is the image at time t, and $\mathbf{C}_t \in R^3$ is the camera translation relative to the body, camera rotation is encoded in $\boldsymbol{\theta}_t$. We assume the shape, $\boldsymbol{\beta}$, is constant. We use a weak-perspective camera projection model [153] where we represent the camera translation in the z axis as the scale parameter. f_{track} is learned by minimizing the following loss for each input video:

$$\mathcal{L}_{\text{track}} = \mathcal{L}_{\text{f}} + \lambda_{\text{r}}\mathcal{L}_{\text{r}} + \lambda_{\text{d}}\mathcal{L}_{\text{d}} + \lambda_{\text{t}}\mathcal{L}_{\text{t}}, \tag{9.11}$$

where $\mathcal{L}_{\text{f}}$, $\mathcal{L}_{\text{r}}$, $\mathcal{L}_{\text{d}}$, and $\mathcal{L}_{\text{t}}$ are the fitting, rendering, data prior, and temporal consistency losses, respectively, and λ_{r}, λ_{d}, and λ_{t} are their weights. We set $\lambda_{\text{r}} = 1, \lambda_{\text{d}} = 0.1$, and $\lambda_{\text{t}} = 0.01$ in our experiments. The overview of our optimization framework is described in Figure 9.5.

L_{f} and $\mathcal{L}_{\text{r}}$ utilize image-based dense UV map predictions [13] which enforce the 3D body fits to better align with the image space silhouettes of the body. Specifically, L_{f} measures the 2D distance between the projected 3D vertex locations and corresponding 2D points in the image:

$$\mathcal{L}_{\text{f}} = \sum_{\mathbf{X} \leftrightarrow \mathbf{x} \in \mathcal{U}} \|\Pi_p \mathbf{X} - \mathbf{x}\|. \tag{9.12}$$

where $\mathcal{U}$ is the set of dense keypoints in the image, $\mathbf{x} \in R^2$ obtained from image-based dense UV map predictions [297], $\mathbf{X}$ are the corresponding 3D vertices, and Π_p is the camera projection which is a function of $\mathbf{C}$. $\mathcal{L}_{\text{r}}$ measures the difference between the rendered and detected UV maps, $\mathbf{y}$:

$$\mathcal{L}_{\text{r}} = \|g(\mathcal{W}^{-1}\mathbf{p}^t, \mathbf{C}_t) - \mathbf{y}\|, \tag{9.13}$$

where $g(\cdot)$ is the differentiable rendering function that renders the UV coordinates from the 3D body model.

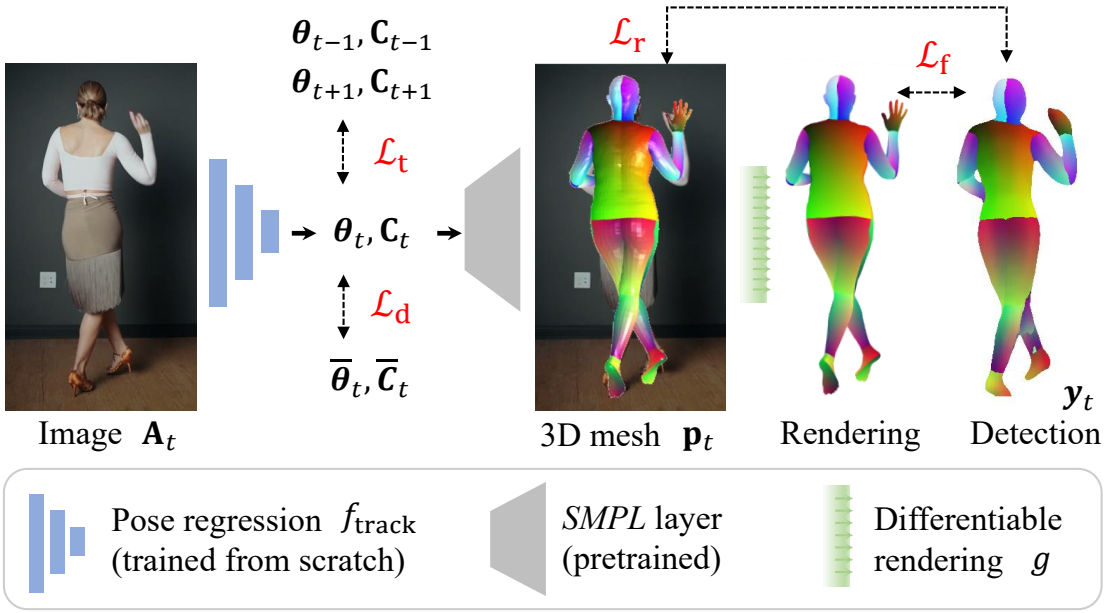


Figure 9.5. The overview of our model-based monocular 3D performance tracking. A regression network predicts the body ($\boldsymbol{\theta}$) and camera ($\mathbf{C}$) pose parameters from a single image. The pretrained SMPL layer [17] decodes the predicted parameters to reconstruct the posed 3D body mesh. We render out the dense IUV coordinates of the mesh using a differentiable rendering layer and train the regression network by enforcing self-consistency between densepose detection and rendered IUV map [13] ($\mathcal{L}_{\mathrm{r}}$ and $\mathcal{L}_{\mathrm{f}}$); and enforcing temporal smoothness ($\mathcal{L}_{\mathrm{t}}$) and data-driven regularization ($\mathbf{L}_{\mathrm{d}}$).

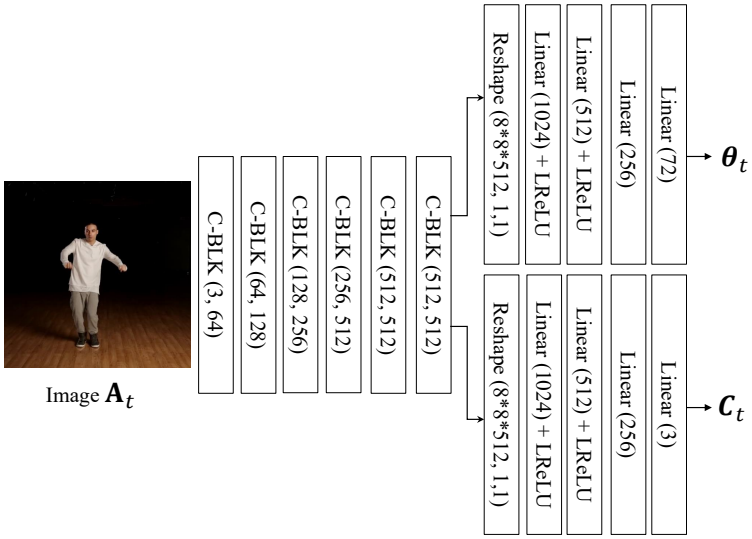


Figure 9.6. Network design for our 3D body and camera pose regression network (f_{track}). The details for C-BLK, D-BLK, Conv, and LReLU are described in Figure 9.7.

$\mathcal{L}_{\text{d}}$ provides the data driven prior on body and camera poses, i.e., $\mathcal{L}_{\text{d}} = \|\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}\| + \|\mathbf{C} - \overline{\mathbf{C}}\|$, where $\overline{\boldsymbol{\theta}}$ and $\overline{\mathbf{C}}$ are the initial body and camera parameters predicted by a state-of-the-art method [293]. $\mathcal{L}_{\text{t}}$ enforces the temporal smoothness over time: $\mathcal{L}_{\text{t}} = \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}\| + \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t+1}\| + \|\mathbf{C}_t - \mathbf{C}_{t-1}\| + \|\mathbf{C}_t - \mathbf{C}_{t+1}\|$.

We enable f_{track} using a convolutional neural network. The details of our network designs are described in Figure 9.6. where it predicts the 3D body $\boldsymbol{\theta}$ and camera $\mathbf{C}$ pose from an image $\mathbf{A}$.

9.2 Network Designs

We learn our motion encoder E_{Δ} and compositional rendering decoders, E_s, E_a using convolutional neural networks. In this section, we provide the implementation details of our network designs.

3D Pose Tracking Network We enable f_{track} using a convolutional neural network. The details of our network designs are described in Figure 9.6. where it predicts the 3D body $\boldsymbol{\theta}$ and camera $\mathbf{C}$ pose from an image $\mathbf{A}$.

3D Motion Encoder Network, E_{Δ}. Figure 9.8 describes the network details for our

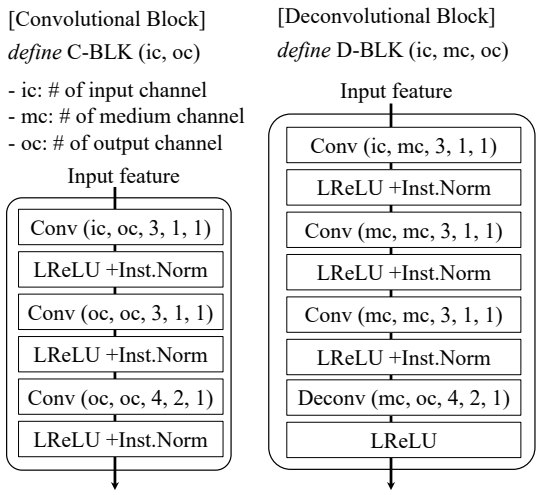


Figure 9.7. Implementation details of our convolutional and deconvolutional blocks. Conv and Deconv denotes convolutional and deconvolutional layers are constructed based on the parameters: number of input channels (ic), number of output channels (oc), filter size, stride, and the size of the zero padding. We set the coefficient of the LeakyReLU (LReLU) to 0.2.

3D motion encoder E_{Δ}. It takes as input 3D surface normal $\mathbf{N}_t$ of the current frame and velocity $\mathbf{V}_t$ for past 10 frames recorded in the UV space of the body and outputs 3D motion descriptors $\mathbf{f}_{3\text{D}}^t$.

Shape Decoder Network, E_s. Figure 9.9 describes the network details for our shape decoder network E_s which takes as input the 3D motion descriptor $\widehat{f}_t$ rendered in the image space and the predicted shape in the previous time instance $\widehat{\mathbf{s}}_{t-1}$, and outputs the person-specific 2D shape $\widehat{\mathbf{s}}_t$ which is composed seven category label maps.

Appearance Decoder Network, E_a. Figure 9.10 describes the network details for our appearance decoder network E_s which takes as input the projected 3D motion descriptor $\widehat{f}_t$ rendered in the image space, predicted shape $\widehat{\mathbf{s}}_t$, and the predicted appearance and surface normal in the previous time instance $\{\widehat{\mathbf{A}}_{t-1}, \widehat{\mathbf{n}}_{t-1}\}$, and outputs the 3D surface normal $\widehat{\mathbf{n}}_t$ and appearance $\widehat{\mathbf{A}}_t$.

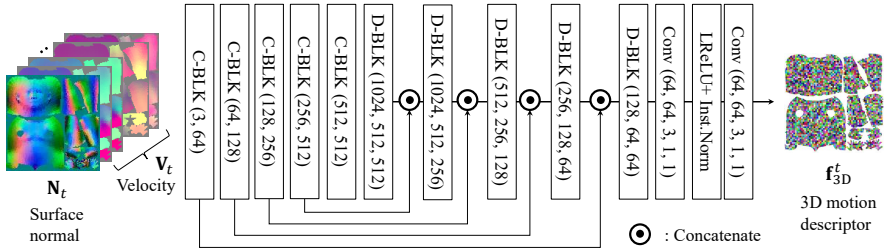


Figure 9.8. Network design for our 3D motion encoder (E_Δ). The details of C-BLK, D-BLK, Conv, and LReLU are described in Figure 9.7.

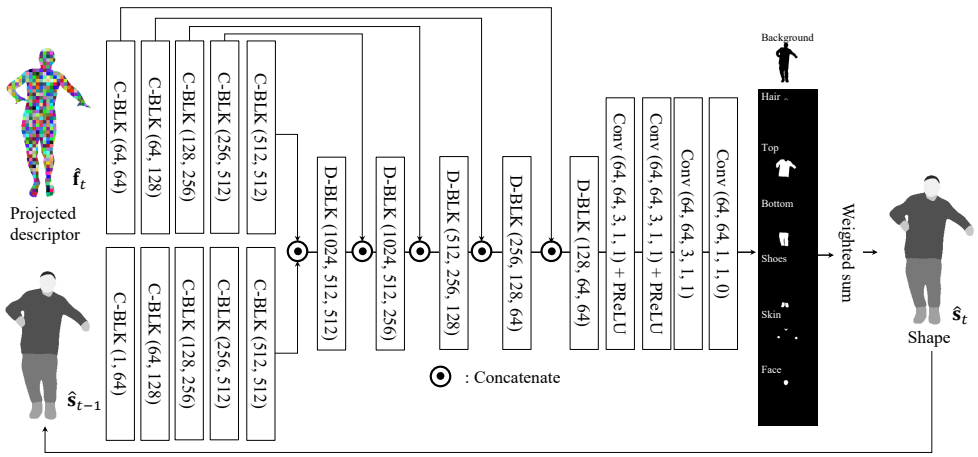


Figure 9.9. Network design for our shape decoder (D_s). The details of C-BLK, D-BLK, Conv, and LReLU are described in Figure 9.7.

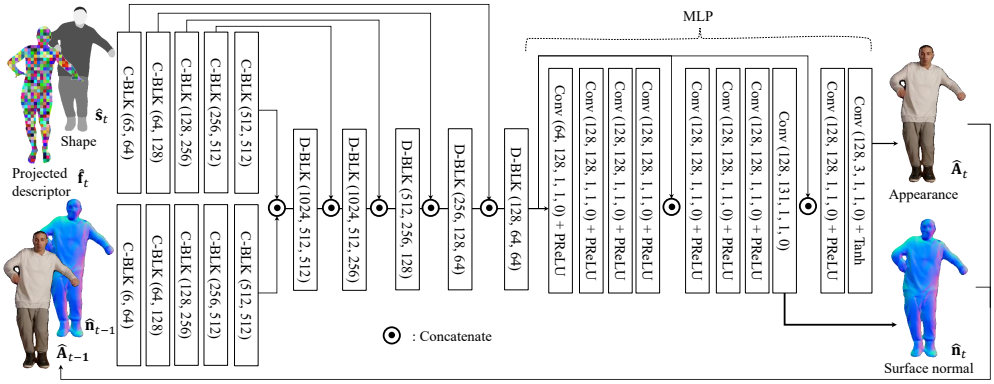


Figure 9.10. Network design for our appearance decoder (D_a). The details of C-BLK, D-BLK, Conv, and LReLU are described in Figure 9.7.

9.3 Experiments

We validate the performance of our method across various examples and perform extensive qualitative and quantitative comparisons with previous work.

Implementation Details. We utilize the Adam optimizer [130] to train our model with a learning rate of 1×10^{-3}. Given an input video ($\sim 10K$ frames), we train our model for roughly 72 hours using 4 NVIDIA V100 GPUs using a batch size of 4. Our motion features are learned from the body surface normals in the current frame and the body surface velocities in the past $t = 10$ frames. The features are recorded in a UV map of size 128×128. We synthesize final renderings and surface normal maps of size 512×512. We implement our model in Pytoch and utilize the Pytorch3D differentiable rendering layers [298]. Details of the network designs and 3D tracking pipeline are given in the supplementary materials.

The coordinate transformation that transports the motion features from the UV space to the image plane, i.e., Π in Equation (9.5), can be implemented either by using image-based dense UV estimates [13] or by directly rendering the UV coordinates of the 3D body fits. To provide a fair comparison with previous work which also utilize dense UV estimates, we use the former option. When demonstrating our method on applications where we do not have corresponding ground truth frames to estimate dense UV maps (e.g., novel viewpoint synthesis), we use the latter option.

Baselines. We compare ours to four prior methods that focused on synthesizing dressed humans in motion. 1) Everybody dance now (EDN) [26] uses image-to-image translation to synthesize human appearance conditioned on 2D keypoints and uses a temporal discriminator to enforce plausible dynamic appearance. 2) Video-to-video translation (V2V) [27] is a sequential video generator that synthesizes high-quality human renderings from 2D keypoints and dense UV maps where the motion is modeled with optical flow in the image space. 3) High-fidelity motion transfer (HFMT) [28] is a compositional recurrent network which predicts plausible motion-dependent shape and appearance from 2D keypoints. 4) Dance in the wild (DIW) [4] synthesizes dynamic appearance of humans based on a motion descriptor composed of time-consecutive 2D keypoints and dense UV maps. We evaluate only on foreground by removing the background synthesized by the methods EDN, V2V, and DIW using a human segmentation method [56].

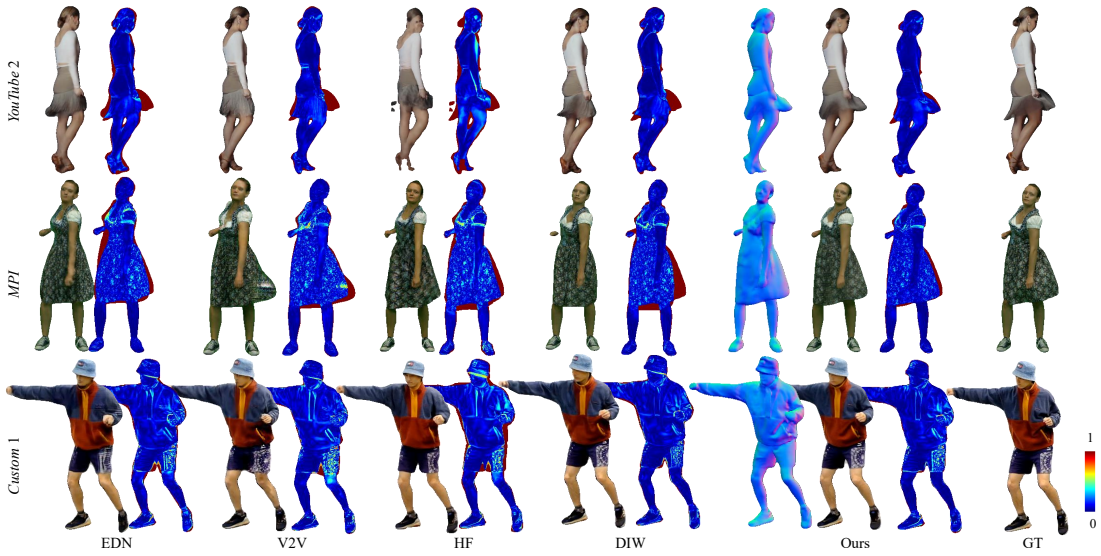


Figure 9.11. We compare our method to several baselines (EDN [26], V2V [27], HFMT [28], DIW [4]) on various sequences. For each example, we show the ground truth (GT) target appearance, the synthesized appearance by each method, and a color map of the error between the two. For our method, we also visualize the predicted surface normal.

HFMT predicts a foreground mask similar to ours. In the supplementary material, we also compare our method to the 3D based approach [222] for neural avatar modeling from a single camera, which explicitly reconstruct the geometry of animatable human.

Datasets. We perform experiments on video sequences that demonstrate a wide range of motion sequences and clothing types, which include non-trivial secondary motion. Specifically, we select three dance videos (e.g., hip-hop and salsa) from YouTube and one sequence from prior work [28] that shows a female subject in a large dress. We also capture two custom sequences showing a male and a female subject respectively performing assorted motions (e.g., walking, running, punching, jumping etc.) including 3D rotations.

Metrics. We measure the quality of the synthesized frames with two metrics: 1) Structure similarity (SSIM) [299] compares the local patterns of pixel intensity in the normalized luminance and contrast space. 2) Perceptual distance (LPIPS) [63]

Method	*YouTube* 1 (6K)	*YouTube* 2 (10K)	*YouTube* 3 (4K)	*MPI* (10K)
EDN [26]	0.954 / 3.06 / 0.356	0.943 / 4.39 / 0.465	0.871 / 6.23 / 0.467	0.824 / 4.59 / 0.287
V2V [27]	0.960 / 2.23 / 0.235	0.958 / 3.33 / 0.405	0.880 / 4.47 / 0.401	0.824 / 3.58 / 0.298
HFMT [28]	0.944 / 4.19 / 0.412	0.923 / 6.63 / 0.775	0.862 / 7.16 / 0.456	0.826 / 5.03 / 0.291
DIW [4]	0.966 / 2.21 / 0.275	0.960 / 3.03 / 0.370	0.894 / 4.69 / 0.396	0.825 / 2.94 / 0.359
Ours	0.973 / 2.01 / 0.240	0.964 / 2.83 / 0.338	0.897 / 4.50 / 0.412	0.825 / 2.82 / 0.203

Custom 1 (15K)	*Custom* 2 (15K)	Avg.
0.916 / 5.26 / 0.450	0.928 / 5.06 / 0.423	0.906 / 4.76 / 0.408
0.935 / 3.52 / 0.306	0.943 / 4.15 / 0.385	0.916 / 3.54 / 0.338
0.905 / 6.24 / 0.321	0.915 / 6.63 / 0.390	0.895 / 5.98 / 0.440
0.939 / 3.23 / 0.304	0.944 / 3.95 / 0.412	0.921 / 3.34 / 0.336
0.942 / 3.12 / 0.279	0.946 / 3.81 / 0.404	0.925 / 3.18 / 0.312

Table 9.1. Quantitative results. The number of training frames in each sequence is given in the top row. The thee numbers are the SSIM (↑), LPIPS (↓)×100, and tLPIPS (↓)×100 metrics, respectively. The red represents the best performer, and the blue second best.

evaluates the cognitive similarity of a synthesized image to ground truth by comparing their perceptual features extracted from a deep neural network. We evaluate the temporal plausibility by comparing the perceptual change across frames [300]: tLPIPS = $\|\text{LPIPS}(s_t, s_{t-1}) - \text{LPIPS}(g_t, g_{t-1})\|$ where s and g are the synthesized and ground truth images.

9.3.1 Evaluation

Comparisons We provide quantitative evaluation in Table 9.1 and show qualitative results in Figure 9.11 (see Supplementary Video). Similar to our method, we train each baseline for roughly 72 hours until convergence. Both qualitative and quantitative results show that sparse 2D keypoint based pose representation used in EDN is not as effective as other baselines or our method. HFMT is successful in modeling dynamic appearance changes for mostly planar motions (i.e., MPI sequence), but shows inferior performance in remaining sequences that involve 3D rotations. This is due to the depth ambiguity inherent in sparse 2D keypoint based representation. While V2V performs well in terms of quantitative numbers, it suffers from significant texture drifting issues as shown in Figure 9.11, second row. We speculate that this is due to the errors in the

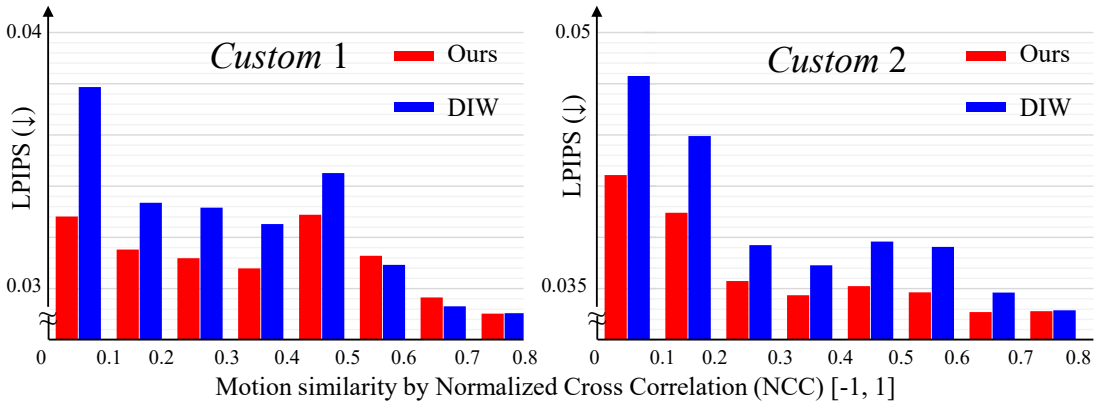


Figure 9.12. Perceptual quality of a synthesized image over motion similarity between training and testing sequences.

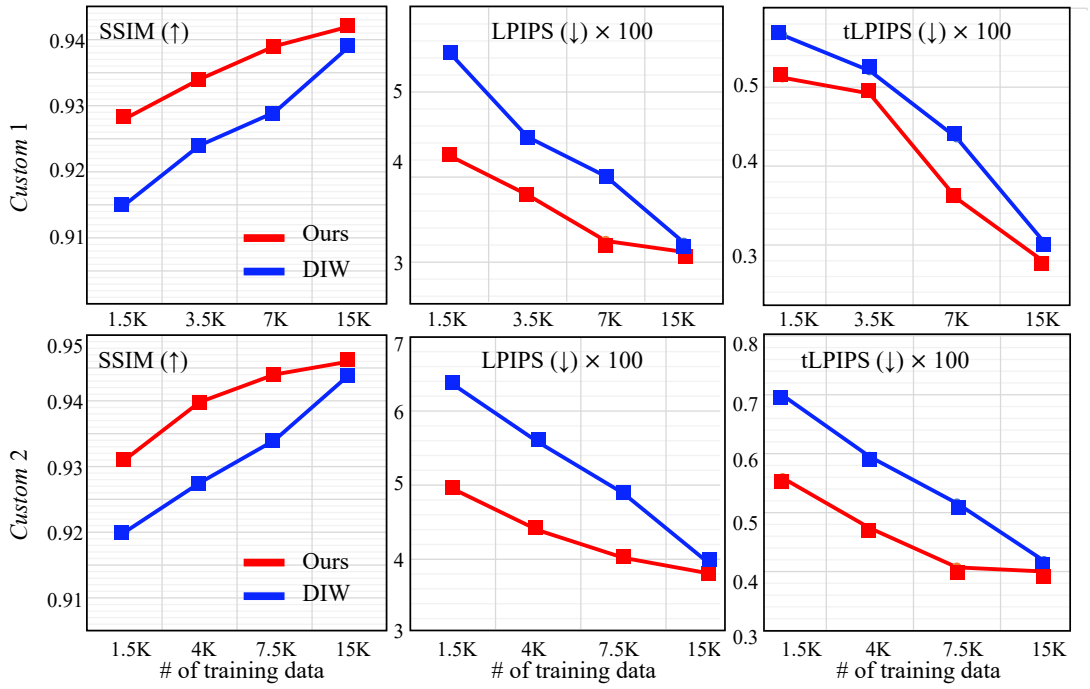


Figure 9.13. Performance depending on the amount of training data.

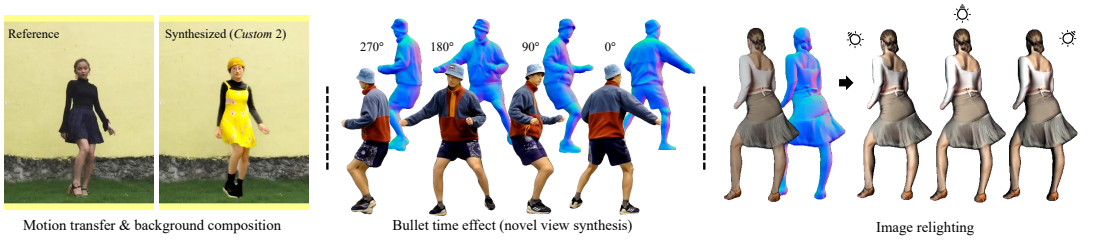


Figure 9.14. Application. Our method enables several applications such as motion transfer with background composition, bullet time effects with novel view synthesis, and image-based relighting with the predicted surface normal.

Method	*YouTube* 1 (0.6K)	*YouTube* 2 (1K)	*YouTube* 3 (0.4K)	*MPI* (1K)
DIW [4]	0.939 / 4.12 / 0.330	0.940 / 5.00 / 0.463	0.869 / 6.75 / 0.513	0.824 / 4.45 / 0.472
Ours	0.949 / 3.36 / 0.457	0.951 / 4.16 / 0.402	0.883 / 5.35 / 0.546	0.824 / 4.09 / 0.327

Custom 1 (1.5K)	*Custom* 2 (1.5K)	Avg.
0.915 / 5.46 / 0.566	0.920 / 6.36 / 0.698	0.901 / 5.36 / 0.507
0.928 / 4.25 / 0.512	0.931 / 4.95 / 0.558	0.911 / 4.36 / 0.467

Table 9.2. We train DIW [4] and our method on a reduced training set (10% of the original training set) and test on the same testing set. The thee numbers in each box represent the SSIM (↑), LPIPS (↓)×100, and tLPIPS (↓)×100 metrics, respectively.

optical flow estimation, especially in case of loose clothing, which is used as a supervisory signal. DIW uses dense UV coordinates to model the dynamic appearance changes of loose garments and is the strongest baseline. While it performs consistently well, we observe that the performance gap between DIW and our method increases for motion segments consisting of 3D rotations. This gap is magnified when the testing motion deviates from the training data. In Figure 9.12, we plot how the perceptual error changes along the motion similarity between the training and testing data which is computed by NCC between two sets of the time-varying 3D meshes similar to Figure 9.4. We observe bigger increase in the error for DIW as testing frames deviate more from the training data.

We next perform further comparisons with DIW evaluating the generalization ability of each approach.

Generalization An effective motion representation that encodes the dynamic appearance change of dressed humans should be discriminative to distinguish all possible deformations induced by a pose transformation given the current state of the body and the garments. In order to compare the discriminative power of our motion descriptor and the dense keypoint based representation proposed by DIW, we evaluate how well each representation generalizes to unseen poses. Specifically, we train each model using only 10% of the original training sequences by subsampling the training frames while ensuring training and testing pose sequences are sufficiently distinct. Considering the reduced amount of data, we limit the training time to 24 hours for both approaches. As shown in Table 9.2, the performance gap between the two methods increases. For the *Custom* 1 and 2 sequences, we further repeat the same experiment using 10%, 25%, and 50% of the original training data as shown in Figure 9.13 where the performance of our method shows slower degradation than that of DIW. These quantitative results as well as visual results provided in the supplementary materials demonstrate the superiority of our 3D motion descriptor in terms of generalizing to novel poses.

Ablation Study Using the *Custom* 2 sequence, we train a variant **w/o 3D motion descriptors** by providing dense uv renderings as input directly to the decoder. We also disable the shape (**w/o shape**) and surface normal (**w/o surface normal**) prediction components. We repeat these trainings with subsampled data (10%). As shown in Table 9.3, the use of 3D motion descriptors and compositional rendering improves the perceptual quality of the synthesized images. The performance gap between our full model and w/o surface normal is larger with limited training data, implying that our multi-task framework helps with generalization. Qualitative results are given in the supplementary materials.

9.3.2 Applications

Our method enables several additional applications as shown in Figure 9.14. Since our method works with 3D body based motion representation, it can be easily used to transfer motion from a source to a target character by simply transferring the joint rotations between the characters. We can also create bullet time effects by creating a target motion sequence by globally rotating the 3D body. Thanks to the surface normal prediction, we can also perform relighting which is otherwise not applicable. Please

Method	Full data (15K)	10% data (1.5K)
w/o shape	0.945 / 4.31 / 0.401	0.929 / 5.28/ 0.565
w/o surface normal	0.945 / 3.89 / 0.418	0.929 / 5.17 / 0.602
w/o 3D motion	0.942 / 4.17 / 0.584	0.928 / 5.43 / 0.760
Full	0.946 / 3.81 / 0.404	0.931 / 4.95 / 0.558

Table 9.3. Ablation study. The three metrics are SSIM (↑), LPIPS (↓)×100, and tLPIPS (↓)×100 respectively. The number in the top row denotes the amount of training data.

refer to the supplementary material for more details and results.

9.3.3 Evaluation for Monocular 3D Pose Tracking

We validate the performance of our 3D pose tracking method by comparing with previous monocular image based (SPIN [147] and SMPLx [292]) and video based (VIBE [293]) 3D body estimation methods.

We use the AIST++ dataset [301] which provides pseudo-ground truth SMPL fits obtained from multiview images. For randomly selected four subjects, we select four viewpoints and two motion styles (600 frames per motion) resulting in 4800 testing frames per subject. Due to the differences in the camera models adopted by each method (*i.e.,* perspective or orthographic cameras), there exist a scale ambiguity between the predictions and the ground truth. Hence, we measure the per-vertex 2D projection error between the ground truth and predicted 3D body model in the image space. We provide quantitative and qualitative results in Table 9.4 and Figure 9.15, respectively. By exploiting both temporal cues and dense keypoint estimates, our method outperforms the previous work.

9.4 Conclusion

We presented a method to render the dynamic appearance of a dressed human given a reference monocular video. Our method utilizes a novel 3D motion descriptor that encodes the time varying appearance of garments to model effects such as secondary

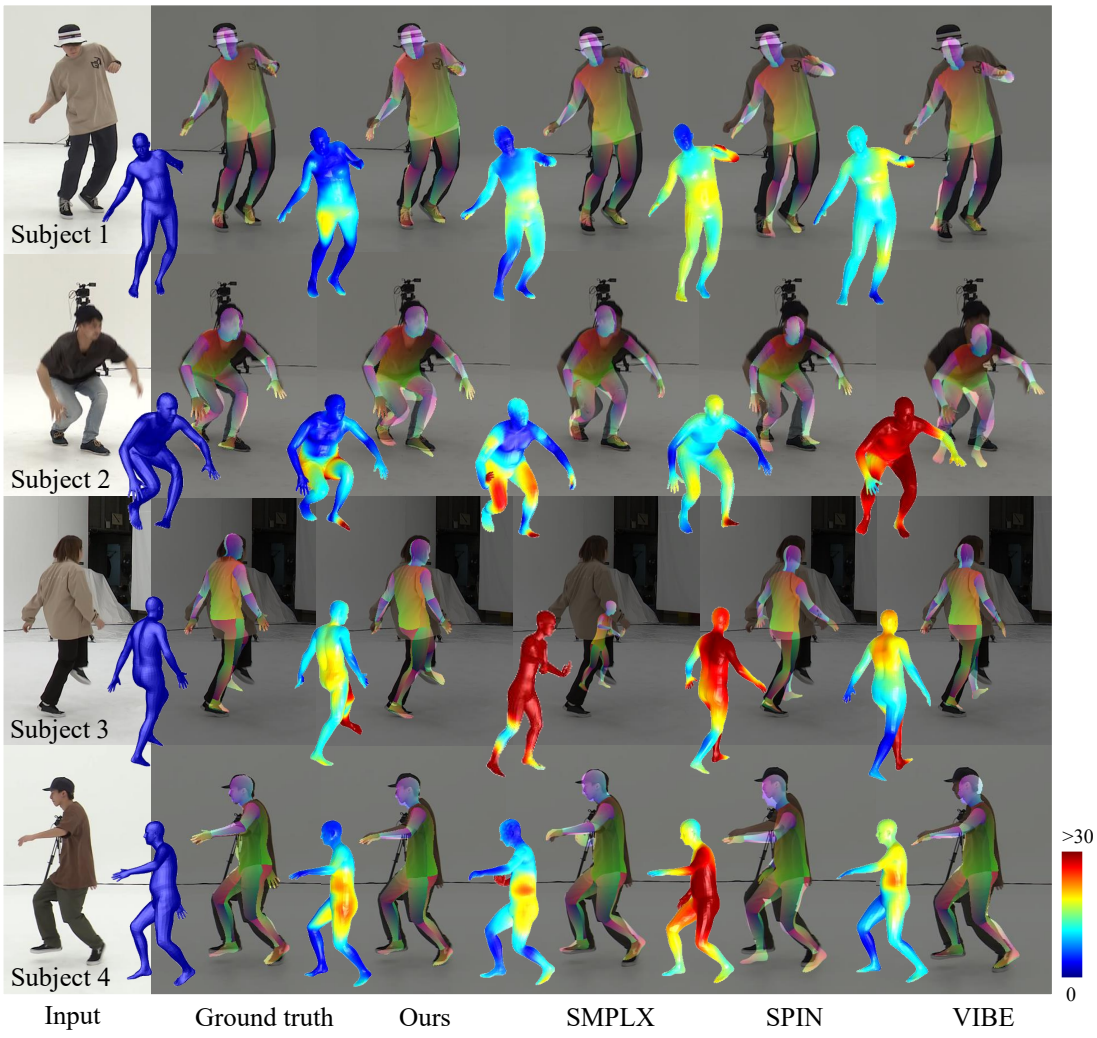


Figure 9.15. We show the 3D body estimates and color coded 2D projection errors of our method and baselines for images of size 512×512.

	Sub.1	Sub.2	Sub.3	Sub.4	Avg.
SPIN [147]	16.5±3.7	22.6±6.6	23.4±6.2	21.5±4.4	21.0±5.2
VIBE [293]	13.9±2.9	12.2±2.8	17.7±5.1	15.5±2.9	14.8±3.4
SMPLx [292]	9.0±1.6	10.2±1.7	16.2±10.2	12.1±4.4	11.9±4.5
Ours	8.3±1.1	8.7±2.0	13.7±3.5	11.3±1.9	10.5±2.1

Table 9.4. We show the mean and std of per-vertex projection error between the ground truth and estimated 3D bodies for images of size 512×512.

motion. Our experiments show that our 3D motion descriptor is effective in modeling complex motion sequences involving 3D rotations. Our descriptor also demonstrates superior discriminator power compared to state-of-the-art alternatives enabling our method to better generalize to novel poses.

While showing impressive results, our method still has limitations. Highly articulated hand regions can appear blurry, hence refining the appearance of such regions with specialized modules is a promising direction. Our current model is subject specific, extending different parts of the model, e.g., 3D motion descriptor learning, to be universal is also an interesting future direction.

Chapter 10

Conclusion

10.1 Summary

In this thesis, we introduce a new method to develop a novel AI model that can reconstruct high-quality 3D human avatars from a single camera by learning from visual data. We address the core challenge of data scarcity problem: there exists no 3D ground truth data to learn that covers diverse human appearance and geometry from our everyday environment. This challenge highly limits the application of the learned AI model to a specific scene and person. To overcome this challenge, we propose a set of self-supervised approaches that can learn a generalizable human visual representation to reconstruct 3D avatars from a single image; to adapt the learned avatar to general scenes; and to render the avatars for diverse people.

Learning to reconstruct 3D avatars from a single camera We presents a method to predict a high-fidelity 3D human avatar from a single image by learning from multi-view data. We build a large corpus of human visual dataset to facilitate high resolution pose- and view-specific appearance of human body expressions. The dataset includes diverse activities of facial expression, body and clothing movement, finger gesture, and for 772 distinctive subjects across gender, ethnicity, age, and physical condition. Given multiview image streams, we reconstruct full-body 3D mesh models including face, hand, body, and gaze where we use semantic trajectory priors to improve the quality and temporal coherence of 3D reconstruction results. The images and 3D reconstruction results are used to train a new monocular variational auto-encoder in a way that reconstructs

high-fidelity 3D human avatar from a single image.

Learning to adapt the learned 3D avatars to general unconstrained scenes We propose a self-supervised algorithm that can adapt the learned 3D avatar to any unconstrained scene beyond the controlled lab environment. Our self-supervised adaptation method leverages the scene-invariant assumption that the position and color of 3D human mesh models over two consecutive frames should not change drastically. This assumption enables us to extract supervision to fine-tune the visual representation from in-the-wild video frames without any 3D ground truth data. The results demonstrate that our method not only improves visual quality of the reconstructed 3D avatars but also the temporal stability of the avatars' animation under the domain changes, e.g, scenes and viewpoint changes.

Learning to render fine-Grained appearance of 3D avatars of diverse people. We introduce a method to synthesize fine-grained appearance of dynamic humans without high-quality geometry. In particular, we employ a person-agnostic undressed 3D body model which are readily obtainable from any unconstrained images using previous monocular 3D pose prediction methods. For example, a generative neural network is specially designed to take as input reference person image and 3D body model from novel body poses and output fine-grained appearance from the novel poses. To enhance the physical plausibility over times, we further embed temporal derivatives of the 3D body models over time on our generative network, which allows this to decode motion-dependent appearance for physically plausible human animation. Since a person-specific geometry is no longer requirement, our method is highly generalizable to diverse people for fine-grained appearance rendering.

10.2 Limitation and Future Works

Our method shows the following limitations.

Lack of full-body plausibility. While our methods improve the overall visual quality of 3D human avatars compared to previous methods, highly articulated body parts, in particular, for face, hair, shoes, and hand, are often appearing blurry or missing without high-frequency details For example, in Fig. 10.1-(Left), the right hand of the rendered women is missing and the texture of the hair and face is more blurry compared

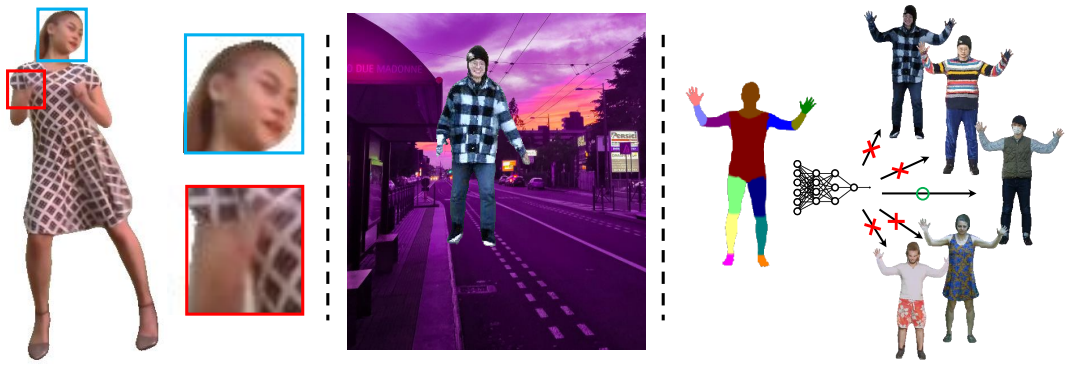


Figure 10.1. Limitations of our method. (Left): Lack of full-body plausibility; (Middle): Lack of scene context; and (Right): Lack of model generalizability.

to the clothing texture. This is mainly due to the fact that the learned single rendering network is highly biased to synthesize clothing texture which is the major component that constitutes the human appearance, and the latent representation for those small body parts are largely removed. To mitigate such model bias, in our future work, we will enable a full-body human rendering machine which will be composed of multiple modular neural networks each of which is specially dedicated for a specific body part, e.g., a face network is dedicated to animate and render only for face.

Lack of scene context. The rendered 3D avatars should contextually make sense, reflecting the state of surrounding environment decided by a number of factors such as 3D scene structure and lighting. However, our rendering machines learned from a video of a specific scene synthesize the appearance of 3D avatars without such context, resulting in mismatch between a novel background scene and the foreground human rendering. For example, in Fig. 10.1-(Middle), the visual statistics of the rendered 3D avatars (foreground) highly deviates from the one of the background scenes, and the scale and location of the 3D avatar is not geometrically plausible, i.e., the person looks like floating on the road. To address this problem, we will introduce a deep blending network that can composite a 3D avatar with any background scenes in a contextually and geometrically plausible way. The network will be designed in a way that extracts the scene intrinsic such as lighting and 3D structure from a single image, and they will be conditioned on a decoder to predict nature appearance of 3D avatars and its ideal location, e.g., x and y position, in the context of the scenes.

Lack of computational efficiency for training. Many of our current AI models are person-specific, i.e., a single neural network is only able to handle one person as shown in Fig. 10.1-(Right), and training a new AI model for another identity requires tremendous computational time such as two or three days. Such computational complexity is a huge barrier for the interaction between the model and end users of many practical applications, e.g., avatar-based social tele-presence, from which people desire prompt outputs. In our future work, we will address this problem by developing a generalizable AI model that can handle multiple people from a single neural network. In particular, we are interested in exploring an efficient transfer learning algorithm such as meta-learning framework [302, 303] which enables fast adaptation of a neural network to any person with small number of data.

References

[1] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016.

[2] Albert Pumarola, Jordi Sanchez, Gary Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3DPeople: Modeling the Geometry of Dressed Humans. In *ICCV*, 2019.

[3] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*, 2019.

[4] Tuanfeng Y. Wang, Duygu Ceylan, Krishna Kumar Singh, and Niloy J. Mitra. Dance in the wild: Monocular human animation with neural dynamic appearance synthesis, 2021.

[5] Anna Shvets. People on a video call. `https://www.pexels.com/photo/people-on-a-video-call-4226140/`, 2020.

[6] Kaiwen Guo, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, et al. The relightables: Volumetric performance capture of humans with realistic relighting. *SIGGRAPH*, 2019.

[7] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei

Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *TPAMI*, 2017.

[8] Hanbyul Joo, Tomas Simon, Mina Cikara, and Yaser Sheikh. Towards social artificial intelligence: Nonverbal social signal prediction in a triadic interaction. In *CVPR*, 2019.

[9] Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J. Black. Dyna: A model of dynamic human shape in motion. *SIGGRAPH*, 2015.

[10] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4d clothing capture and retargeting. *SIGGRAPH*, 2017.

[11] D. Knossow, R. Ronfard, and R. Horaud. Human motion tracking with a kinematic parameterization of extremal contours. *IJCV*, 2008.

[12] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *CVPR*, 2015.

[13] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018.

[14] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3d human sensing in natural environments. *TPAMI*, 2014.

[15] Jae Shin Yoon, Takaaki Shiratori, Shoou-I Yu, and Hyun Soo Park. Self-supervised adaptation of high-fidelity face models for monocular performance tracking. In *CVPR*, 2019.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[17] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *SIGGRAPH*, 2015.

[18] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. Neural re-rendering of humans from a single image. In *ECCV*, 2020.

[19] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017.

[20] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *CVPR*, 2017.

[21] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

[22] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2020.

[23] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

[24] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. 2020.

[25] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017.

[26] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *ICCV*, 2019.

[27] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018.

[28] Moritz Kappel, Vladislav Golyanik, Mohamed Elgharib, Jann-Ole Henningson, Hans-Peter Seidel, Susana Castillo, Christian Theobalt, and Marcus Magnor. High-fidelity neural human motion transfer from monocular video. In *CVPR*, 2021.

[29] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, and Peng Liu. A high-resolution spontaneous 3d dynamic facial expression database. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*. IEEE, 2013.

[30] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 2013.

[31] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 2009.

[32] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *TPAMI*, 2003.

[33] Shiyang Cheng, Irene Kotsia, Maja Pantic, and Stefanos Zafeiriou. 4dfab: A large scale 4d database for facial expression analysis and biometric applications. In *CVPR*, 2018.

[34] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3D face model for pose and illumination invariant face recognition. *International Conference on Advanced Video and Signal Based Surveillance*, 2009.

[35] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3D morphable models. *IJCV*, 2017.

[36] Haotian Yang, Hao Zhu, Yanru Wang, Mingkai Huang, Qiu Shen, Ruigang Yang, and Xun Cao. Facescape: a large-scale high quality 3d face dataset and detailed riggable 3d face prediction. In *CVPR*, 2020.

[37] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *SIGGRAPH*, 2014.

[38] A. Wetzler, R. Slossberg, and R. Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. In *BMVC*, 2016.

[39] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T-K. Kim. Big hand 2.2m benchmark: Hand pose data set and state of the art analysis. In *CVPR*, 2017.

[40] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *ICCV*, 2017.

[41] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. A hand pose tracking benchmark from stereo matching. In *ICIP*, 2017.

[42] Jimei Yang Bryan Russel Max Argus Christian Zimmermann, Duygu Ceylan and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 2019.

[43] S.I. Park and J.K. Hodgins. Capturing and animating skin deformation in human motion. *SIGGRAPH*, 2006.

[44] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 2010.

[45] Tomas Simon, Hanbyul Joo, Iain A Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[46] Chao Zhang, Sergi Pujades, Michael J. Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3d scan sequences. In *CVPR*, 2017.

[47] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018.

[48] Timo von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. Human pose estimation from video and imus. *PAMI*, 2016.

[49] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *CVPR*, 2017.

[50] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[51] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[52] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPRW*, 2014.

[53] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[54] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[55] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017.

[56] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *ECCV*, 2018.

[57] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021.

[58] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural animation and reenactment of human actor videos. *SIGGRAPH*, 2019.

[59] Jack Valmadre and Simon Lucey. General trajectory prior for non-rigid reconstruction. In *CVPR*, 2012.

[60] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. Deep appearance models for face rendering. *SIGGRAPH*, 2018.

[61] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *SIGGRAPH*, 2019.

[62] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. *SIGGRAPH*, 2021.

[63] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[64] Zhixuan Yu, Jae Shin Yoon, In Kyu Lee, Prashanth Venkatesh, Jaesik Park, Jihun Yu, and Hyun Soo Park. Humbi: A large multiview dataset of human body expression. 2020.

[65] Jae Shin Yoon, Zhixuan Yu, Jaesik Park, and Hyun Soo Park. Humbi: A large multiview dataset of human body expressions and benchmark challenge. *TPAMI*, 2021.

[66] Jae Shin Yoon, Ziwei Li, and Hyun Soo Park. 3d semantic trajectory reconstruction from 3d pixel continuum. *CVPR*, 2017.

[67] Orazio Gallo Hyun Soo Park Jae Shin Yoon, Kihwan Kim and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. 2020.

[68] Vladislav Golyanik Kripasindhu Sarkar Hyun Soo Park Jae Shin Yoon, Lingjie Liu and Christian Theobalt. Pose-guided human animation from a single image in the wild. In *CVPR*, 2021.

[69] Tuanfeng Y. Wang Jingwan Lu Jimei Yang Zhixin Shu Jae Shin Yoon, Duygu Ceylan and Hyun Soo Park. Learning motion-dependent appearance for high-fidelity rendering of dynamic humans from a single camera. In *CVPR*, 2022.

[70] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012.

[71] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT press Cambridge, 2016.

[72] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[73] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on automatic face and gesture recognition (FGR06)*. IEEE, 2006.

[74] Wojciech Sankowski, Piotr Stefan Nowak, and Paweł Krotewicz. Multimodal biometric database dmcsv1 of 3d face and hand scans. In *2015 22nd International Conference Mixed Design of Integrated Circuits & Systems (MIXDES)*. IEEE, 2015.

[75] Kieron Messer, Jiri Matas, Josef Kittler, Juergen Luettin, and Gilbert Maitre. Xm2vtsdb: The extended m2vts database. In *International Conference on Audio and Video-based Biometric Person Authentication*, 1999.

[76] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[77] Ryan White, Keenan Crane, and David A Forsyth. Capturing and animating occluded cloth. In *SIGGRAPH*, 2007.

[78] Derek Bradley, Tiberiu Popa, Alla Sheffer, Wolfgang Heidrich, and Tamy Boubekeur. Markerless garment capture. In *TOG*, 2008.

[79] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. *ECCV*, 2020.

[80] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *CVPR*, 2020.

[81] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *CVPR*, 2016.

[82] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *CVPR*, 2017.

[83] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *ICCVW*, 2013.

[84] Erjin Zhou, Haoqiang Fan, Zhimin Cao, Yuning Jiang, and Qi Yin. Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *ICCVW*, 2013.

[85] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. *TPAMI*, 2013.

[86] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.

[87] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *SIGGRAPH*, 2017.

[88] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[89] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[90] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI*, 2013.

[91] Mixamo. `https://www.mixamo.com/#/`.

[92] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *CVPR*, 2019.

[93] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019.

[94] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*, 2018.

[95] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *ICCV*, 2017.

[96] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, 2010.

[97] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, 2012.

[98] Susanna Ricco and Carlo Tomasi. Video motion for every visible point. In *ICCV*, 2013.

[99] Ehsan Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. In *CVPR*, 2009.

[100] Lorenzo Torresani and Christoph Bregler. Space-time tracking. In *ECCV*, 2002.

[101] Shankar Rao, Roberto Tron, R. Vidal, and Yi Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *PAMI*, 2010.

[102] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *ICCV*, 2009.

[103] B. Taylor, A. Ayvaci, A. Ravichandran, and S. Soatto. Semantic video segmentation from occlusion relations within a convex optimization framework. In *CVPR Workshop*, 2013.

[104] Abhijit Kundu, Yin Li, Frank Daellert, Fuxin Li, and James M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *ECCV*, 2014.

[105] Abhijit Kundu, Vibhav Vineet, and Vladlen Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, 2016.

[106] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR*, 1999.

[107] L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *CVPR*, 2001.

[108] Appu Shaji, Aydin Varol, Lorenzo Torresani, and Pascal Fua. Simulataneous point matching and 3D deformable surface reconstruction. In *CVPR*, 2010.

[109] Jingyu Yan and Marc Pollefeys. Automatic kinematic chain building from feature trajectories of articulated objects. In *CVPR*, 2006.

[110] M. Salzmann, J. Pilet, S. Ilic, and P. Fua. Surface deformation models for nonrigid 3D shape recovery. *PAMI*, 2007.

[111] Jonathan Taylor, Allan D. Jepson, and Kiriakos N. Kutulakos. Non-rigid structure from locally-rigid motion. In *CVPR*, 2010.

[112] I. Akhter, Y. Sheikh, and S. Khan. In defense of orthonormality constraints for nonrigid structure from motion. In *CVPR*, 2009.

[113] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. In *CVPR*, 2012.

[114] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In *NeurIPS*, 2008.

[115] Hyun Soo Park, Takaaki Shiratori, Iain Matthews, and Yaser Sheikh. 3D reconstruction of a moving point from a series of 2D projections. In *ECCV*, 2010.

[116] Shai Avidan and Amnon Shashua. Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *PAMI*, 2000.

[117] Jeremy Yirmeyahu Kaminski and Mina Teicher. A general framework for trajectory triangulation. *Journal of Mathematical Imaging and Vision*, 2004.

[118] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2D image motion. In *ECCV*, 2000.

[119] Hyun Soo Park and Yaser Sheikh. 3d reconstruction of a smooth articulated trajectory from a monocular image sequence. In *ICCV*, 2011.

[120] Ijaz Akhter, Tomas Simon, Sohaib Khan, Iain Matthews, and Yaser Sheikh. Bilinear spatiotemporal basis models. *SIGGRAPH*, 2012.

[121] H. Joo, H. S. Park, and Y. Sheikh. Map visibility estimation for large-scale dynamic 3d reconstruction. In *CVPR*, 2014.

[122] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *ICCV*, 2015.

[123] A. Del Bue, X. Lladó, and L. Agapito. Segmentation of rigid motion from non-rigid 2d trajectories. *Pattern Recognition and Image Analysis*, 2007.

[124] C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3d reconstruction of dynamic scenes. In *NeurIPS*, 2014.

[125] Katerina Fragkiadaki, Marta Salas, Pablo Arbelaez, and Jitendra Malik. Grouping-based low-rank trajectory completion and 3d reconstruction. In *NeurIPS*, 2014.

[126] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *CVIU*.

[127] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. *TPAMI*.

[128] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999.

[129] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3D face morphable models "in-the-wild". In *CVPR*, 2017.

[130] Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

[131] Sami Romdhani and Thomas Vetter. Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *CVPR*, 2005.

[132] Joseph Roth, Yiying Tong, and Xiaoming Liu. Adaptive 3D face reconstruction from unconstrained photo collections. In *CVPR*, 2016.

[133] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. FaceWarehouse: A 3D facial expression database for visual computing. *TVCG*.

[134] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *ICCV*, 2017.

[135] László A Jeni, Jeffrey F Cohn, and Takeo Kanade. Dense 3D face alignment from 2D video for real-time use. *Image Vision Comput.*, 2017.

[136] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li. Face alignment in full pose range: A 3D total solution. *TPAMI*, 2019.

[137] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. In *CVPR*, 2017.

[138] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3D morphable models with a very deep neural network. In *CVPR*, 2017.

[139] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz. In *CVPR*, 2018.

[140] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3D face reconstruction and dense alignment with position map regression network. In *ECCV*, 2018.

[141] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophyics*, 1973.

[142] Alessio Del Bue, Xavier Llad, and Lourdes Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *CVPR*, 2006.

[143] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *TPAMI*, 2008.

[144] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. *ECCV*, 2020.

[145] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020.

[146] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *ICCV*, 2019.

[147] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.

[148] Camillo J. Taylor. Reconstruction of articulated objects from point correspondences in a single image. In *CVPR*, 2000.

[149] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *ECCV*, 2012.

[150] M. A. Brubaker, D. J. Fleet, and A. Hertzmann. Physics-based person tracking using simplified lower-body dynamics. In *CVPR*, 2007.

[151] X. Wei and J. Chai. Videomocap: Modeling physically realistic human motion from monocular video sequences. *SIGGRAPH*, 2010.

[152] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *ECCV*, 2016.

[153] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[154] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *ICCV*, 2009.

[155] Zorah Lahner, Daniel Cremers, and Tony Tung. Deepwrinkles: Accurate and realistic clothing modeling. In *ECCV*, 2018.

[156] Tao Yu, Zerong Zheng, Yuan Zhong, Jianhui Zhao, Qionghai Dai, Gerard Pons-Moll, and Yebin Liu. Simulcap: Single-view human performance capture with cloth simulation. In *CVPR*, 2019.

[157] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *TOG*, 2016.

[158] Genzhi Ye, Yebin Liu, Nils Hasler, Xiangyang Ji, Qionghai Dai, and Christian Theobalt. Performance Capture of Interacting Characters with Handheld Kinects. In *ECCV*, 2012.

[159] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *CVPR*, 2018.

[160] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, and Jonathan Taylor. Fusion4d: real-time performance capture of challenging scenes. *TOG*, 2016.

[161] S. Olsen and A. Bartoli. Using priors for improving generalization in non-rigid structure-from-motion. In *BMVC*, 2007.

[162] Adrien Bartoli, Vincent Gay-Bellile, Umberto Castellani, Julien Peyras, Søren Olsen, and Patrick Sayd. Coarse-to-fine low-rank structure-from-motion. In *CVPR*, 2008.

[163] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, 2019.

[164] Igor Santesteban, Miguel A Otaduy, and Dan Casas. Learning-based animation of clothing for virtual try-on. In *Computer Graphics Forum*, 2019.

[165] Hadi Fadaifard and George Wolberg. Image warping for retargeting garments among arbitrary poses. *The Visual Computer: International Journal of Computer Graphics*, 2013.

[166] Jiaxiang Shang, Tianwei Shen, Shiwei Li, Lei Zhou, Mingmin Zhen, Tian Fang, and Long Quan. Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency. *ECCV*, 2020.

[167] Sicheng Xu, Jiaolong Yang, Dong Chen, Fang Wen, Yu Deng, Yunde Jia, and Xin Tong. Deep 3d portrait from a single image. In *CVPR*, 2020.

[168] Yajing Chen, Fanzi Wu, Zeyu Wang, Yibing Song, Yonggen Ling, and Linchao Bao. Self-supervised learning of detailed 3d face reconstruction. *IEEE Transactions on Image Processing*, 29, 2020.

[169] Elisa Ricci, Wanli Ouyang, Xiaogang Wang, Nicu Sebe, et al. Monocular depth estimation using multi-scale continuous crfs as sequential deep networks. *TPAMI*, 2018.

[170] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[171] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[172] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, pages 740–756, 2016.

[173] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*, 2017.

[174] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body and hands in the wild. In *CVPR*, 2019.

[175] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, 2019.

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

[177] Feitong Tan, Hao Zhu, Zhaopeng Cui, Siyu Zhu, Marc Pollefeys, and Ping Tan. Self-supervised human depth estimation from monocular videos. In *CVPR*, 2020.

[178] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014.

[179] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. *NeurIPS*, 2017.

[180] Lingbo Yang, Pan Wang, Chang Liu, Zhanning Gao, Peiran Ren, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Xiansheng Hua, and Wen Gao. Towards fine-grained human pose transfer with detail replenishing network. *TIP*, 2021.

[181] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *CVPR*, 2019.

[182] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, 2018.

[183] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *CVPR*, 2018.

[184] Hao Tang, Dan Xu, Gaowen Liu, Wei Wang, Nicu Sebe, and Yan Yan. Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *ICMM*, 2019.

[185] Liqian Ma, Zhe Lin, Connelly Barnes, Alexei A Efros, and Jingwan Lu. Unselfie: Translating selfies to neutral-pose portraits in the wild. In *ECCV*. Springer, 2020.

[186] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *ECCV*, 2018.

[187] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018.

[188] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In *CVPR*, 2020.

[189] Sijie Song, Wei Zhang, Jiaying Liu, and Tao Mei. Unsupervised person image generation with semantic parsing transformation. In *CVPR*, 2019.

[190] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *CVPR*, 2019.

[191] Assaf Neuberger, Eran Borenstein, Bar Hilleli, Eduard Oks, and Sharon Alpert. Image based virtual try-on network from unpaired data. In *CVPR*, 2020.

[192] Xu Chen, Jie Song, and Otmar Hilliges. Unpaired pose guided human image generation. In *CVPRW*, 2019.

[193] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, 2018.

[194] Badour AlBahar and Jia-Bin Huang. Guided image-to-image translation with bi-directional feature transformation. In *ICCV*, 2019.

[195] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. Clothflow: A flow-based model for clothed person generation. In *ICCV*, 2019.

[196] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image generation. In *CVPR*, 2020.

[197] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *ECCV*, 2018.

[198] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018.

[199] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019.

[200] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara Berg. Dance dance generation: Motion transfer for internet videos. In *ICCVW*, 2019.

[201] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *ECCV*, 2018.

[202] Tsun-Hsuan Wang, Yen-Chi Cheng, Chieh Hubert Lin, Hwann-Tzong Chen, and Min Sun. Point-to-point video generation. In *ICCV*, 2019.

[203] Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhoefer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, and Christian Theobalt. Neural human video rendering by learning dynamic textures and rendering-to-video translation. *TVCG*, 2020.

[204] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. *NeurIPS*, 2019.

[205] Sergey Prokudin, Michael J Black, and Javier Romero. Smplpix: Neural avatars from 3d human models. In *WACV*, 2021.

[206] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, et al. Textured neural avatars. In *CVPR*, 2019.

[207] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural rendering and reenactment of human actor videos. *SIGGRAPH*, 2019.

[208] Amit Raj, Julian Tanke, James Hays, Minh Vo, Carsten Stoll, and Christoph Lassner. Anr: Articulated neural rendering for virtual avatars. In *CVPR, year=2021.*

[209] Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabian Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason Saragih. Driving-signal aware full-body avatars. *SIGGRAPH*, 2021.

[210] Donglai Xiang, Fabian Andres Prada, Timur Bagautdinov, Weipeng Xu, Yuan Dong, He Wen, Jessica Hodgins, and Chenglei Wu. Explicit clothing modeling for an animatable full-body avatar. *arXiv preprint arXiv:2106.14879*, 2021.

[211] Meng Zhang, Tuanfeng Y. Wang, Duygu Ceylan, and Niloy J. Mitra. Dynamic neural garments. *SIGGRAPH*, 2021.

[212] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[213] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. *ICCV*, 2020.

[214] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*, 2021.

[215] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. *ICCV*, 2021.

[216] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv*, 2021.

[217] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

[218] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. *ICCV*, 2021.

[219] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *SIGGRAPH Asia*, 2021.

[220] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021.

[221] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Animatable neural radiance fields for human body modeling. In *ICCV*, 2021.

[222] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, and Huchuan Lu. Animatable neural radiance fields from monocular rgb video. *ICCV*, 2021.

[223] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. 2021.

[224] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. *ECCV*, 2020.

[225] Oran Gafni, Oron Ashual, and Lior Wolf. Single-shot freestyle dance reenactment. In *CVPR*, 2021.

[226] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided human image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12135–12144, 2019.

[227] Badour AlBahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. Pose with style: Detail-preserving pose-guided image synthesis with conditional stylegan. *SIGGRAPH Asia*, 2021.

[228] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[229] Matthieu Armando, Jean-Sébastien Franco, and Edmond Boyer. Adaptive mesh texture for multi-view appearance modeling. In *3DV*, 2018.

[230] Thabo Beeler, Bernd Bickel, Paul Beardsley, Bob Sumner, and Markus Gross. High-quality single-shot capture of facial geometry. *SIGGRAPH*, 2010.

[231] Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *SIGGRAPH*, 2005.

[232] A. Laurentini. The visual hull concept for silhouette-based image understanding. *TPAMI*, 1994.

[233] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017.

[234] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

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

[236] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, 2015.

[237] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, 2007.

[238] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017.

[239] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, 2017.

[240] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

[241] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Interactive foreground extraction using iterated graph cuts. *SIGGRAPH*, 2012.

[242] Hao Tang, Dan Xu, Yan Yan, Jason J Corso, Philip HS Torr, and Nicu Sebe. Multi-channel attention selection gans for guided image-to-image translation. *CVPR*, 2019.

[243] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[244] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019.

[245] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018.

[246] K. E. Ozden, K. Cornelis, L. V. Eychen, and L. V. Gool. Reconstructing 3D trajectories of independently moving objects using generic constraints. *CVIU*, 2004.

[247] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from Internet photo collections. *IJCV*, 2008.

[248] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001.

[249] Andrew Delong, Anton Osokin, Hossam N. Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *IJCV*, 2012.

[250] JiaWang Bian, Wen-Yan Lin, Yasuyuki Matsushita, Sai-Kit Yeung, Tan Dat Nguyen, and Ming-Ming Cheng. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In *CVPR*, 2017.

[251] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[252] Jae Shin Yoon, Francois Rameau, Junsik Kim, Seokju Lee, Seunghak Shin, and In So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017.

[253] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[254] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[255] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 2016.

[256] Ran Tao, Efstratios Gavves, and Arnold W. M. Smeulders. Siamese instance search for tracking. In *CVPR*, 2016.

[257] Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *SIGGRAPH*, 2012.

[258] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *CVPR*, 2020.

[259] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *CVPR*, 2020.

[260] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *CVPR*, 2020.

[261] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, 2019.

[262] Jack Valmadre and Simon Lucey. General trajectory prior for non-rigid reconstruction. In *CVPR*, 2012.

[263] C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *CVPR*, 2011.

[264] Luca Ballan, Gabriel J. Brostow, Jens Puwein, and Marc Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *SIGGRAPH*, 2010.

[265] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders:feature learning by inpainting. In *CVPR*, 2016.

[266] Zhaoyang Lv, Kihwan Kim, Alejandro Troccoli, Deqing Sun, James M Rehg, and Jan Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In *ECCV*, 2018.

[267] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[268] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, 2019.

[269] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016.

[270] Qi Zhang, Li Xu, and Jiaya Jia. 100+ times faster weighted median filter (wmf). In *CVPR*, 2014.

[271] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, 2019.

[272] Fangchang Mal and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *ICRA*, 2018.

[273] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *SIGGRAPH*, 2013.

[274] Takeo Igarashi, Tomer Moscovich, and John F Hughes. As-rigid-as-possible shape manipulation. *SIGGRAPH*, 2005.

[275] Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 2011.

[276] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[277] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *ECCV*, 2018.

[278] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *Arxiv*, 2014.

[279] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[280] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 1951.

[281] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *CVPR*, 2019.

[282] Valentin Gabeur, Jean-Sébastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. In *ICCV*, 2019.

[283] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *CVPR*, 2019.

[284] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019.

[285] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *Arxiv*, 2014.

[286] Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *CVPR*, 2017.

[287] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019.

[288] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *SIGGRAPH Asia*, 2020.

[289] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *CVPR*, 2018.

[290] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[291] J. P. Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH*, page 165–172, 2000.

[292] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *ECCV*, 2020.

[293] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020.

[294] Eakta Jain, Yaser Sheikh, Moshe Mahler, and Jessica Hodgins. Augmenting Hand Animation with Three-dimensional Secondary Motion. In MZoran Popovic and Miguel Otaduy, editors, *SCA*, 2010.

[295] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In *CVPR*, June 2021.

[296] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[297] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, 2019.

[298] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *Arxiv*, 2020.

[299] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004.

[300] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *SIGGRAPH*, 2020.

[301] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++.

[302] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. Metaavatar: Learning animatable clothed human models from few depth images. *NeurIPS*, 34, 2021.

[303] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *CVPR*, 2021.