Reconstructing and Rendering People from Photos and Videos in the Wild

Chung-Yi Weng

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

Brian Curless, Chair

Ira Kemelmacher-Shlizerman, Chair

Richard Szeliski

Program Authorized to Offer Degree:

Computer Science & Engineering

University of Washington

Abstract

Reconstructing and Rendering People from Photos and Videos
in the Wild

Chung-Yi Weng

Co-Chairs of the Supervisory Committee:
Brian Curless
Computer Science & Engineering

Ira Kemelmacher-Shlizerman
Computer Science & Engineering

Reconstructing and producing photorealistic renderings of dynamic humans from RGB images has long been considered a holy grail in the fields of computer vision and graphics. Such a capability would open up a wide range of possibilities for applications in areas such as virtual and augmented reality, teleconferencing, and the entertainment industry. Despite more than 25 years of research and development, the problem remains challenging, primarily due to difficulties posed by inherent 3D-to-2D ambiguity, highly dynamic motions, appearance variance, and non-rigid deformation. Moreover, the high cost of the technology has also been a major barrier to widespread adoption, as the reconstruction pipelines often rely on calibrated multi-camera systems and are typically only found in professional studios.

In this thesis, I address the challenge of reconstructing and rendering high-quality dynamic humans using unstructured data in the wild, such as photos from the internet or YouTube videos. The goal is to make this expensive technology more accessible to amateur artists and even the general public, democratizing its use beyond just movie studios. To begin, I provide a review of the literature on this long-established problem, starting with the seminal work of Kanade et al. in 1997 and tracing the evolution of the technology through advances in image-based rendering, surface reconstruction, and more recently, modern deep neural networks. Then I present three novel

approaches for tackling this problem, each designed to work with different types of source material, including monocular videos, personal photo collections, and single photographs. Through these approaches, my research enables a range of new applications.

My proposed first approach, Photo Wake-Up, allows for creating 3D human animations viewable on AR devices like HoloLens using only single images. The second method, known as HumanNeRF, enables free-viewpoint rendering of moving persons from a YouTube video. Finally, I present PersonNeRF, an approach that is capable of reconstructing a person, including tennis superstars like Roger Federer, from photo collections, enabling rendering with arbitrary combinations of their viewpoints, appearances, and body poses. In the final section, I discuss the open problems that still exist in this field, as well as how this technology will potentially shape our future world.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I want to express my gratitude to my amazing advisors, Brian Curless and Ira Kemelmacher-Shlizerman, for their enduring support throughout this journey. Their thoughtful advice and constructive criticism have helped shape my ideas and projects; their belief in my abilities has reminded me of my initial enthusiasm for this enchanting subject; their genuine care and concern have made all the difference during times of uncertainty and self-doubt. I am truly thankful for and cherish every moment we have shared.

I wish to thank Rick Szeliski for his valuable feedback and unwavering commitment to joining me on this journey. Since the early days of my research career, his publications have been a constant source of inspiration, shaping my perspectives and driving my passion. It is both an honor and a privilege to have him on my committee. I wish to thank Tivon Rice for providing his unique viewpoint as an artist for my dissertation. His exceptional works have inspired me to approach my research through the lens of storytelling and connect it to the things I truly care about.

I would like to thank my close collaborators: Jon Barron and Pratul Srinivasan. Jon's problem-solving approach, characterized by a harmonious combination of simplicity, elegance, and effectiveness, has established a golden standard for me. Pratul has played a pivotal role in my research. His expertise, insights, and dedication have been invaluable, providing guidance and shaping the outcome. Without their advice, my work would not have reached its current level of depth and quality.

I am thankful for those people who helped or advised me during my internships at Meta and Google: Michael F. Cohen, Johannes Kopf, Kevin Matzen, Andrew Denyes, Rohit Pandey, Christian Hane, Sofien Bouaziz, and Sean Fanello. Their involvement made the experience unforgettable and valuable for shaping my future career trajectory.

I am grateful to meet numerous researchers, colleagues, and postdoctoral fellows who made my time incredibly enjoyable – Steve Seitz, Adriana Schulz, Linda Shapiro, Paul Debevec, Qi Shan, Supasorn Suwajanakorn, Ricardo Martin-Brualla, Aditya Sankar, Edward Zhang, Shu Liang, Soumyadip Sengupta, Konstantinos Rematas, Haisen Zhao, Chenming Wu, Aleksander Hołyński, Edward Zhang, Xuan Luo, Keunhong Park, Jeong Joon Park, Liang Luo, Ming Liu, Annie Ross, Manaswi Saha, Emily Furst, Zuoming Shi, Adam Fishman, Alice Gao, Nikita Haduong, Vivek Jayaram, Teerapat Jenrungrot, Ben Jones, Johanna Karras, Benlin Liu, Jingwei Ma, Yuxuan Mei, James Noeckel, Roy Or-El, Mengyi Shan, Meng-Li Shih, Isaac Tian, Aaron Walsman, Xiaojuan Wang, Yifan Wang, Kuo-Hao Zeng, and Luyang Zhu.

This thesis would not be possible without my family's love and sacrifice – my father, mother, and sister. Last but not least, I want to thank my two lovely kids - Aaron and Althea - for choosing me as their father and entering my world. Their presence has brought exhaustion but also immense joy. Indeed, there have been moments when I couldn't even understand how I managed to overcome the challenges of raising two kids while pursuing a Ph.D. However, as I complete this journey and find myself able to write down these acknowledgments, it is solely because of my wife. Her unwavering love and support throughout these years have made it possible. I wish to dedicate this thesis to her, with my utmost admiration and deepest appreciation.

DEDICATION

to my dear wife, Yu-Ju Chu

Chapter 1

INTRODUCTION: UNVEILING NEW DIMENSIONS FOR VISUAL STORYTELLING

Storytelling holds a natural power that goes beyond its narrative form. Since the earliest stages of our lives, storytelling sparks our imagination, shapes our perceptions, and teaches us valuable life lessons. In the meantime, the format of storytelling has kept evolving, and people use different media to share their stories. From ancient oral traditions, literature, and film, to modern digital platforms, each format has its own unique strengths and characteristics, and storytellers choose the medium that best suits their message, audience, and creative vision.

In particular, visual storytelling is considered one of the most impactful media, because of its ability to cross language barriers and effectively communicate with a wide-ranging audience. This form of visual storytelling can be demonstrated in various ways, including illustrations, artworks, photography, films, and videos, allowing for engaging audiences from diverse cultural backgrounds without regard to their age or level of education.

Nevertheless, when it comes to visual storytelling in its traditional forms, such as photography or video, there exists a prevalent limitation: they are either static or constrained to two-dimensional space. This inherent constraint hinders the ability to fully capture the narrative environments, engage the audience in interactive ways, and ultimately deliver truly immersive experiences.

Over the years, computer scientists, particularly those in the fields of computer vision and graphics, have played a crucial role in pushing the boundaries of visual storytelling. Their primary goal has been to remove the limitations of traditional visual storytelling by introducing new dimensions, in particular those of **time** and **space**.

This thesis traverses a significant part of the journey toward uncovering new dimensions in visual storytelling. Its primary objective is to reconstruct humans from traditional media used

Figure 1.1: Three projects that lift the visual storytelling to 3D. (a) "New Dimensions in Testimony" [167] create an interactive 3D holographic interview experience; (b) "Starline" [83] aims to achieve an immersive remote video communication; (c) creating the bullet-time scene [181] in the movie "The Matrix" typically requires multiple slow-motion cameras placed around the action in a circle. *Photo credits to University of Southern California, Google, and New World Designs.*

for storytelling, such as paintings, images, or videos, ultimately democratizing this technology and making it accessible to general users. The aim is to enable the telling of stories in a fully immersive, three-dimensional, interactive, and animated format.

1.1 Lifting to 3D

Introducing a new spatial dimension to traditional visual storytelling generally involves lifting from a two-dimensional space to a three-dimensional world. The motivation behind this stems from the desire to create a more engaging narrative experience. This shift opens up new possibilities for perceiving the environment, evoking stronger emotional responses, and allowing for a more immersive connection between the viewer and the subject.

In the project called "New Dimensions in Testimony" [167] proposed at the University of Southern California, computer scientists have created interactive 3D holographic representations of Holocaust survivors (Fig 1.1-(a)). It enables the survivors' stories to be preserved and shared in a compelling and immersive manner, creating a more personal and meaningful connection for users. Another remarkable breakthrough in this field is the "Starline" project [83] developed by Google,

which utilizes a combination of advanced hardware and software to achieve an immersive remote video communication experience (Fig 1.1-(b)). It involves using specialized displays, cameras, and depth sensors arranged in a dedicated setup. The technology captures a person's appearance from multiple perspectives, reconstructing a 3D model of the subject in real-time.

In fact, this remarkable technological advancement can be attributed to an active field known as "free-viewpoint rendering," which has been actively researched in the communities of computer vision and graphics for over 20 years. Dating back to 1999, the movie "The Matrix" used the technology to create a slowed-down, highly stylized action sequence. It involves freezing the action while the camera appears to move around the scene, providing a 360-degree view of a moment frozen in time. This effect demonstrated the capability of creating highly memorable visual experiences and became iconic and widely recognized due to its innovative and striking visual presentation.

Yet, when we delve into the inner working flow of this technology, tremendous effort has been required to achieve a successful showcase. Multiple slow-motion cameras were strategically placed around the action in a circle, simultaneously capturing the scene from different angles (Fig 1.1-(c)). During post-production, the individual frames captured by the cameras were blended together to create a seamless, continuous shot. As a result, the technology has primarily been restricted to professional studios, posing a significant challenge for its accessibility to amateur artists or general users.

To overcome this challenge, I introduce HumanNeRF [178], which removes the need for dense camera views in free-viewpoint rendering and challenges the belief that such technology is only accessible at a high cost. This approach only takes as input a single monocular video (e.g., a YouTube video) and empowers users to freeze the video at any given frame and generate renderings of the subject from entirely different camera angles, including the ability to create a complete 360-degree camera path for that specific frame and body pose.

Figure 1.2: Image animation introduces time dimension on static photographs or paintings. (a) Chuang et al [24] animates pictures with "stochastic motion textures"; the idea was revisited by Holynski et al. [60] with "deep textures"; (c) Averbuch-Elor [7] brings portraits into life.

1.2 Image Animation

On the other hand, media such as art paintings or photographs, in their static format, also have limitations when it comes to storytelling. They freeze a particular scene or composition, which can restrict the ability to convey dynamic narratives that involve the subject movement and present a story in an interactive format. While they can still evoke emotions and spark the viewer's imagination, these static formats can be further complemented and enriched by introducing a time dimension, such as incorporating animation into their presentation.

The problem of animating static photographs has been studied in the field of computer graphics. In 2005, Chuang et al. [24] pioneered the technique of animating pictures by breaking them down into separate layers. They synthesized "stochastic motion textures," which are time-varying 2D displacement maps, and applied them to each layer to create subtle animated motions that mimic natural forces like wind (Fig. 1.2-(a)). This idea was recently revisited by Holynski et al. [60] with their approach called "deep textures," which draws inspiration from modern deep neural networks (Fig. 1.2-(b)). They trained a deep network to encode motion priors from videos of natural scenes. When animating a photo, they encoded pixels as deep features, warped these features using the

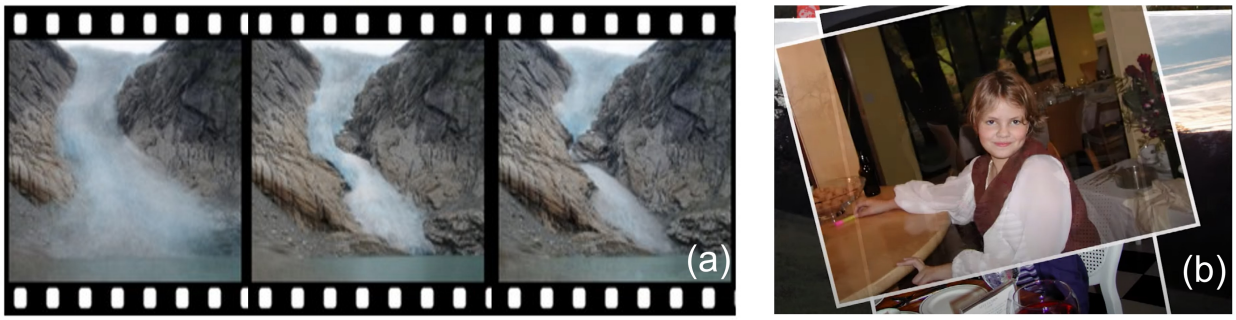


Figure 1.3: Time-lapse animation visualizes the change across static pictures, highlighting its temporal progression. (a) By mining internet photos, Martin-Brualla et al.[101] document the evolution of locations such as the retreat of the glacier. (b) "Face Movie" introduced by Kemelmacher-Shlizerman et al. [75] and Google animates how a person ages across tens of years.

learned motion priors, and trained another neural network to decode the deformed feature map, resulting in a rendered animated scene. Apart from natural scenes, researchers have also developed methods to animate animals [189], portrait photos [7] (Fig. 1.2-(c)), and even human bodies [61, 9].

My proposed approach, Photo Wake-Up [176], extends this line of research by introducing animation into static photographs. Different from previous methods, the approach specifically targets the reconstruction of human subjects in full three-dimensional form. This allows for the creation of human animations that can be experienced on AR devices such as Microsoft HoloLens. With this technology, viewers can interact with the story or the rendered subject in the real world, adding a new level of engagement and immersion.

1.3 Time-lapse Animation

An alternative method of storytelling through animations derived from static photographs is achieved by visualizing the changes across photos. This approach involves presenting a series of photographs that capture different moments or stages of a subject, highlighting its temporal progression and evolution.

Computer researchers, such as Martin-Brualla et al. [101], have successfully applied computa-

tional approaches to expand the scope of this method to a planetary scale. They create animations that document the evolution of various locations worldwide, such as the gradual retreat of the Briksdalsbreen Glacier in Norway, by mining images on the Internet (Fig. 1.3-(a)). Furthermore, Kemelmacher-Shlizerman et al. introduced "Exploring Photobios" [75], which leverages personal photo collections to generate engaging face movies that illustrate how a person ages across tens of years (Fig. 1.3-(b)).

PersonNeRF [179], my solution for personalized human reconstructions, draws inspiration from these previous works. It reconstructs people, including iconic athletes like Roger Federer, from photos spanning many years and enables rendering with arbitrary combinations of their viewpoints, appearances, and body pose. Our approach not only lifts the subjects into 3D by rendering them from arbitrary viewing angles with their distinct body poses, but also allows for time-lapse animation by visualizing their appearance change over time.

1.4 Overview

In the rest of the thesis, Chapter 2 begins with an in-depth exploration of human reconstructions from RGB images. This research area has been an active topic in the fields of computer vision and graphics since the seminal work of "Virtualized Reality" by Kanade et al. in 1997. I delve into the rich history of this problem, highlighting a variety of approaches that have paved the way for further advancements. Furthermore, I explain how my own approaches have been influenced by these previous works, as well as discuss the challenges that need to be addressed.

In Chapter 3, I present my approach, Photo Wake-Up, which allows for creating 3D human animations on AR devices like HoloLens using only single images. Chapter 4 is all about HumanNeRF, a free-viewpoint rendering method that enables the rendering of moving persons viewed from arbitrary cameras, taking only a YouTube video as input. PersonNeRF, the follow-up work of HumanNeRF, is detailed in Chapter 4. It is an approach that is capable of reconstructing subjects from their photo collections across the years, enabling rendering with arbitrary combinations of their viewpoints, appearances, and body poses.

HumanNeRF breaks free from the constraints of monocular videos, granting users the freedom

to select any desired camera perspective and lifting storytelling to immersive three-dimensional experiences, which typically only existed in professional studios previously. Photo Wake-Up brings life into human characters from single photographs, seamlessly integrating them into the real world with the help of AR devices and presenting a brave new way for the audience to interact and engage with art paintings. PersonNeRF takes as input photo collections, and enables not only personalized reconstructions but also time-lapse animations, allowing for rendering the subjects from arbitrary viewpoints while also showcasing their transformations throughout the years.

Finally, I conclude in Chapter 6 by highlighting the open problems that remain and offering insights into the potential impact of these technologies on our future world.

Chapter 2

25 YEARS ON: A SURVEY OF RENDERING DYNAMIC HUMANS FROM RGB IMAGES

2.1 Introduction

Over more than the last two decades, rendering dynamic scenes with continuous viewpoint controls from limited view observations has been a central challenge in computer vision and graphics. Reconstructing moving humans is of particular interest due to its diverse range of applications in telepresence [83, 118], movie production [181], and sports game broadcasting [63]. Solving the problem involves the tasks of capturing, representing, and rendering dynamic subjects.

In the early days, a huge amount of effort was devoted to building capture studios and reconstructing scenes from multiple viewpoints with calibrated cameras. The early exploration had a strong connection in the literature with scene reconstruction [142, 82] and image-based rendering [84, 46]. The proposed approaches were purely data-driven [69, 198, 153], model-based [18], or a combination of both [171].

Recently, researchers have investigated using deep neural networks to improve rendering quality and reduce the necessity of dense view observations. The approaches were initially based in the 2D image space, directly learning a mapping function between 2D body poses and synthesized images, similar to image-to-image translation [64]; later operated in the 3D domain, leveraging the ideas of neural renderer and/or neural representation to guarantee view consistency and enhance visual quality, named after "neural rendering" [162, 163].

In this chapter, I will survey the progress in the area of rendering dynamic humans, beginning from the early works on surface reconstruction (Sec. 2.2) to the modern techniques of neural rendering (Sec. 2.3). Lastly, I will explain how my proposed methods draw inspiration from prior works and discuss the specific challenges I have overcome in order to build particular applications

(Sec. 2.4).

2.2 The Early Years

Generating novel views from observed viewpoints plays a central role in the applications of free-viewpoint rendering. Most of the early works involve rendering static scenes. The seminal works, Light Field Rendering [84] and the Lumigraph [46], achieved photorealistic results by utilizing a vast number of images captured from densely sampled viewpoints. However, the requirement of dense sampling within the viewing space poses a challenge for these methods when attempting to extend them to dynamic scenes.

In order to address this issue, early attempts at modeling dynamic humans employed data-driven approaches that utilized purposefully-designed 3D representations to reduce the need for densely-sampled cameras. Alternatively, model-based methods employed a person-specific rigged mesh to track body motions and rebuild texture accordingly, effectively leveraging human priors. Furthermore, a combination of both approaches was developed to take advantage of the strengths of these two paradigms. In addition, the recently introduced parametric statistical human model, SMPL, [95] inspired a large number of works in the field of 3D human pose and shape estimation, demonstrating an alternative perspective to capture humans from monocular images.

2.2.1 Data-driven Approach

The earliest attempt at capturing moving persons was proposed in 1997 by Kanade et al. [69] who introduced the term "virtualized reality" and popularized capturing people from multiple points of view to the community. They built a 5-meter dome that was equipped with 51 cameras with 512×512 resolution. To reconstruct the subject, they generated a dense depth map for each camera and converted it into a 3D triangle mesh that passed through all the points on that map. In order to account for occlusion, they first eliminated triangles that contained large depth discontinuities along any of their edges, and then filled in the resulting "holes" with the meshes rendered from the other observed camera views. While the result is promising, the visual quality of the output was still limited by factors such as low resolution, incorrect depth estimations, and the presence of

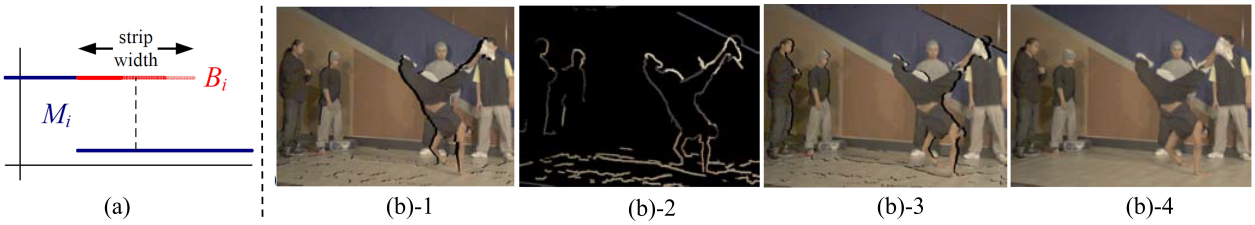


Figure 2.1: (a) The proposed two-layer representation in [198]; (b) The rendering process: (b)-1 rendered main layer from one view; (b)-2 rendered boundary layer; (b)-3 rendered main layer from the other view; (b)-4 final blended result. *Photo credits to [198]*

artifacts around object boundaries.

The follow-up work by Zitnick et al. [198] provided a more cost-effective solution but enabled more realistic rendering. They used 8 high-resolution cameras (1024×768) arranged along a 1D arc spanning about 30 degrees, with the goal of achieving high-quality view interpolation between these views. To this end, they first utilized a segmentation-based method to recover smooth and high-quality depth maps. They then proposed a two-layer representation (Fig. 2.1-(a)) that included a main layer and a boundary layer, with the matte of boundary regions automatically extracted to enhance the quality around object boundaries. At rendering time, they warped all layers from the reference views and blended them with the opacity values from the extracted matte (Fig. 2.1-(b)).

Instead of relying on depth maps, Starck et al. [153] reconstructed a subject's surface from a 3D volume. In their studio, they spaced 8 cameras with 1920×1080 resolution around a circle of 8 meters in diameter. To capture a subject's surface, they first used silhouettes from multiple camera views to derive the visual hull, which identifies infeasible space. They then determined contours on the surface by performing feature matching between the cameras. Once this was completed, they extracted the surface by applying GraphCut on a discrete volumetric graph. In this graph, the source nodes corresponded to the infeasible regions, the sink nodes corresponded to the extracted surface contours, and the edges represented appearance consistency between the camera views. To recover texture, they composited image colors from different cameras and applied multiple-

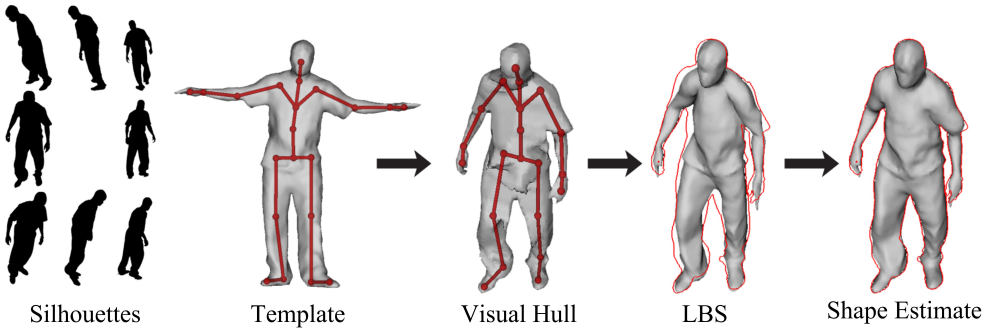


Figure 2.2: [171] starts with multi-view silhouettes and a rigged template mesh. At every frame, it fits the skeleton to the visual hull, deforms the template via LBS, and adjusts the deformed template to fit the silhouettes. *Photo credits to [171].*

resolution blending on the surface to construct a single seamless texture.

2.2.2 Model-based Method

Model-based techniques, on the other hand, leverage the model prior, fitting a humanoid model to multiple-view images. To this end, Carranza et al. [18] employed a manually designed parametric body model, with shape determined by an initialization pose, to track a subject's motion via maximizing the overlap between projected model silhouettes and camera silhouettes. They rebuilt the texture by back-projecting image colors on the mesh vertices, determining vertex visibility, and blending color values with vertex weights calculated based on the angle between the surface normal and the viewing direction.

2.2.3 A Combination of Model-free and Model-based Algorithm

While data-driven approaches without prior allow capturing arbitrary deforming surfaces, they do not always reconstruct the correct topology and may fail to maintain vertex correspondence between frames. On the other hand, model-based techniques ensure the correct topology for human subjects but may not be able to capture fine details. Vlastic et al. [171] combined both methods

to extract a mesh animation with full correspondence from multi-view video recordings while enabling the recovery of detailed surface deformations. Specifically, the process began with a rigged template mesh that contains a skeleton. At every frame, they fitted the skeleton to the visual hull computed from multi-view silhouettes, reposed the template via LBS (linear blending skinning), and finally, deformed the reposed template to fit the silhouettes. We refer the readers to Fig. 2.2 for the pipeline illustration.

Most follow-up works, particularly those that focus solely on surface reconstruction [31, 43, 188, 50, 51], typically adhered to this paradigm, which involved first using a pre-rigged personalized body model, followed by motion tracking and surface deformation that align with the image observations.

2.2.4 SMPL-based Reconstruction

Another notable research direction in reconstructing humans is those methods built upon SMPL [95], a 3D human model with pose and shape controls learned from 3D scans. Although there existed relevant statistical models [5] such as SCAPE [6] when SMPL was introduced, SMPL quickly drew people's attention because it is open-source, easy to operate, and compatible with existing rendering pipelines.

The essence of this line of work is to regress an input image to SMPL body pose and shape coefficient via numeric optimization [14] or deep neural networks [70, 80]. By enforcing temporal consistency in an autoregressive manner, they can be further extended to videos [72, 79]. Most methods do not reconstruct appearances, hence not suitable for novel view synthesis.

In particular, the work presented by Alldieck et al. [3] describes a method to obtain a 3D human avatar from a single monocular video. It shares similarities with the work proposed by Vlasic et al. [171] that uses a template body model to track human motions and deforms the human mesh to align with observed subject silhouettes. They took SMPL as the template model and tracked human motions using the method proposed in [14]. To create texture, they back-projected image colors to the deformed mesh via orthographic projection and computed approximated median values. The result is a deformed SMPL model equipped with a texture map ready for pose controls. This

method requires the subject moves in A-pose, though, and can not capture appearance dynamics as the person is moving.

2.2.5 Reconstruction from RGBD images

Another area of research involves the integration of RGB cameras and depth sensors to capture and reconstruct humans in motion, a technique often referred to as volumetric capture. The cutting-edge systems in this field [26] leverage a combination of RGB stereo, IR stereo, and Shape from Silhouette methodologies to achieve exceptional reconstruction quality. Furthermore, Guo et al. [48] have gone a step further by showcasing the re-illumination capabilities of reconstructed body meshes, utilizing programmable LED lights to generate spherical color gradient illumination patterns that enable the reconstruction of full reflectance maps. However, these advancements are typically limited to high-end studios equipped with an array of expensive equipment, including dense RGB cameras, depth sensors, and LED lights.

In addition to relying on large capture systems, researchers have also explored the use of readily available commodity RGBD cameras for human reconstruction. Notably, Newcombe et al. introduced DynamicFusion [113], a groundbreaking approach that progressively constructs the surface of the human body by establishing correspondence from depth maps to a canonical model, all with the use of just a single RGBD camera. This pioneering work has inspired subsequent studies, such as DoubleFusion [191], which additionally incorporates the SMPL model as a human prior to enhance reconstruction quality, or Fushion4D [36], which utilizes sparse RGBD cameras but achieves comparable quality to that obtained from large capture systems. However, this line of methods typically operates within indoor environments and relies on depth sensor data as input, which limits their scalability to images or videos captured in more diverse or uncontrolled settings.

2.3 The Era of Deep Neural Networks

The recent paradigm shift in computer science is a result of the new renaissance of deep neural networks. The area of human rendering has no exceptions. The early exploration is around conditioning a neural network with 2D body poses and synthesizing the corresponding images. These

approaches found their limits in maintaining view consistency when rendering in 3D due to the lack of 3D reasoning. People get around the issue in primarily two ways: (1) utilizing traditional 3D capture pipelines to model 3D scenes and render the scene to create 2D feature maps or images (with artifacts, most likely), and further using neural networks as a "re-renderer" to decode the features, remove artifacts, or enhance details; (2) directly representing 3D scenes using neural networks and, combined with a differentiable renderer, optimizing the visual quality in an end-to-end manner.

2.3.1 2D Mapping Function

Inspired by the success of deep neural networks on the image-to-image translation task [64], people formulate the problem of human synthesis in a similar way. It began with an application of reposing persons from a single image [97] where it took as input an image of a subject and a target 2D body pose map and trained a network to synthesize the subject wearing the clothing in the input image but being retargeted using the target pose. Balakrishnan et al. [9] enhanced the quality by introducing a modular pipeline where they separated a scene into different body parts and layers and trained a UNet to refine each layer individually. Moreover, the image translation techniques were further extended to human video synthesis by enforcing temporal consistency [173, 20].

However, since these approaches are operated directly within the 2D image domain, they are unable to maintain view consistency when utilized for novel view synthesis due to insufficient 3D reasoning.

2.3.2 Neural Renderer

The neural renderer, or "neural rerendering" [108] utilizes established 3D representations such as sprites, point clouds, meshes, and so on to model a scene in the 3D domain. When projected on two-dimensional image planes, the outcome often contains visual distortions due to imperfect reconstructions. A deep neural network is learned as a "re-renderer" to enhance the final outcome. The process usually involves capturing the scene in a classical pipeline. In the paper titled "LookinGood [102]", the authors demonstrate an example where they utilize a volumetric capture

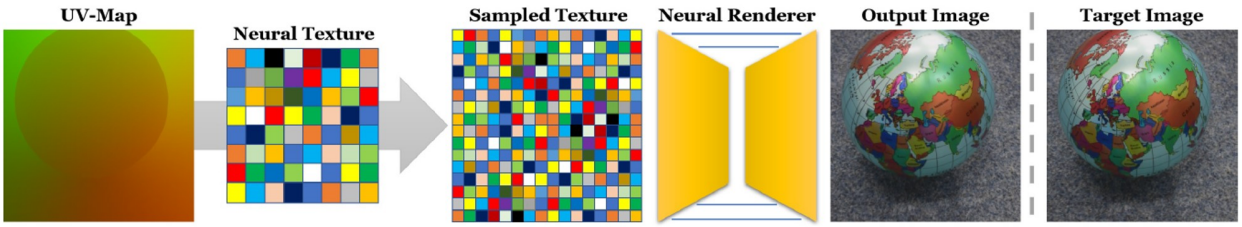


Figure 2.3: An overview of Deferred Neural Rendering [164]. Given an object with a valid UV-map, they associated optimizable features to the map, and used the standard graphics pipeline to render a screen-space feature map. They then trained a UNet to convert the feature map to an output that is visually similar to the target image. *Photo credits to [164].*

system to reconstruct point clouds of performers. They then employ a UNet to remove any visual distortions or artifacts and to elevate the overall quality of the renderings in 2D image space.

As a variant of this concept, Thies et al. proposed a method known as "Deferred Neural Rendering" [164], where they employ a mesh with a valid UV map as the 3D representation, optimize the associated features on the texture map, and train a neural network to decode the rendered feature map (Fig. 2.3). This approach of reinterpreting 2D features can be extended beyond the use of a texture map as a proxy. In the paper Neural Body, Peng and co-authors [126] attach optimizable features to the SMPL mesh vertices and train an MLP-based neural renderer to produce images for unseen views.

In this scenario, the process of 3D reasoning is accomplished through the application of the chosen scene representations that operate in 3D space. The neural networks, on the other hand, are solely responsible for refining or decoding the projected outcome. This approach not only simplifies the task complexity for the networks but also ensures that the resulting views are approximately consistent.

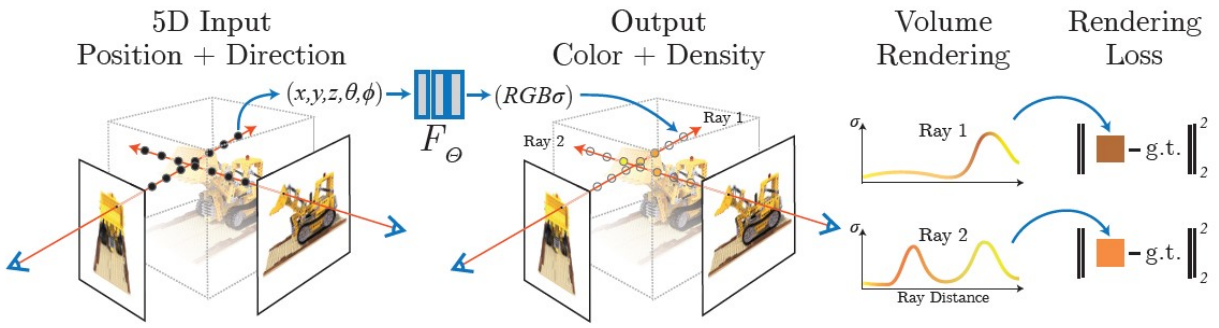


Figure 2.4: An overview of NeRF [110]. NeRF represents a view-dependent volume as a MLP and synthesizes images by sampling and querying from the MLP 5D coordinates (location and viewing direction) along camera rays. The MLP produces a color and volume density. The method then involves volume rendering techniques to composite these values into an image. This rendering function is differentiable, so it can optimize the neural scene representation end-to-end by minimizing the residual between synthesized and ground truth. *Photo credits to [110].*

2.3.3 Neural Representation

Representing a scene with neural networks has emerged to be a powerful tool for solving the problem of novel view synthesis due to the fact that it is robust to arbitrary topologies and, when combined with a differentiable renderer, it can directly optimize the visual quality in an end-to-end manner, avoiding the pitfall of accumulated errors of individual components in a traditional pipeline.

Lombardi et al. in their work "Neural Volumes" [94] directly optimized a $RGB\alpha$ volume in order to reconstruct dynamic scenes from multiple viewpoints. This approach proves to be effective when optimizing visual quality end-to-end. However, their results are not of the highest quality due to limitations in volume resolution. This issue was subsequently addressed by Mildenhall and colleagues in their groundbreaking work known as NeRF [110]. They utilized an MLP to represent the volume. When combined with positional encoding, NeRF was capable of producing nearly photorealistic results that accurately capture view-dependent effects in the context of novel view

synthesis for static scenes (Fig. 2.4).

Neural representations, particularly those based on MLPs, have also proven to be effective in describing scene geometry. In DeepSDF [120] Park et al. optimized an MLP for approximating a signed distance function that implicitly encodes an object's surface. The idea was leveraged by Saito et al. in their work PIFu [136] to predict a textured 3D human mesh from a single image. They used a UNet to extract from image pixel-aligned features, which are subsequently fed into MLPs to predict a subject's geometry as well as appearance. They trained on synthetic 3D human data but applied the learned model to real images. PiFU has sparked a series of research breakthroughs, with the goal of reconstructing a highly detailed human mesh from a single image [137, 62, 53, 195, 185, 4]. In addition, PIFu was further extended to the application of monocular performance capture [86] but the visual quality still remains limited due to the inherited domain gap that leads to quality degradation.

2.4 Inspirations and Challenges

My proposed methods take inspiration from the aforementioned previous works, and in the meantime they also address the unique challenges that arise when reconstructing from unstructured data captured in real-world settings.

The work of Photo Wake-Up [176] utilized the SMPL model [95] to approximate the surface geometry and skinning weights, which were then deformed to accurately fit the subject. This approach allowed for the creation of 3D animatable body meshes from a single image, resulting in realistic and convincing animations. HumanNeRF [178] and PersonNeRF [179] build upon Neural Radiance Fields [110], specifically adapting it to accommodate dynamic human scenes and eliminating the need for dense camera view observations to only single monocular videos or just a handful of images. These advancements have significantly broadened the scope of possibilities in capturing and reconstructing human subjects in a more flexible and accurate manner and enabled a variety of applications.

Chapter 3

PHOTO WAKE-UP: 3D CHARACTER ANIMATION FROM A SINGLE PHOTO

Figure 3.1: Given a single photo as input (far left), we create a 3D animatable version of the subject, which can now walk towards the viewer (middle). The 3D result can be experienced in augmented reality (right); in the result above the user has virtually hung the artwork with a HoloLens headset and can watch the character run out of the painting from different views. *Photo credits to wikiart.org*

This chapter presents the collaborative research project with Brian Curless and Ira Kemelmacher-Shlizerman. The findings from this work were initially published in CVPR 2019 [176]. The subsequent analysis and comparisons to related studies in this chapter are based on the prevailing state-of-the-art during that time.

3.1 Introduction

We propose to "wake up a photo" by bringing the foreground character to life, so that it can be animated in 3D and emerge from the photo. Related to our application are cinemagraphs and

GIFs[1] where a small motion is introduced to a photo to visualize dominant dynamic areas. Unlike a cinemagraph, which is a 2D experience created from video, our method takes a single photo as input and results in a fully 3D experience. The output animation can be played as a video, viewed interactively on a monitor, and as an augmented or virtual reality experience, where a user with a headset can enjoy the central figure of a photo coming out into the real world.

A central challenge in delivering a compelling experience is to have the reconstructed subject closely match the silhouette of the clothed person in the photo, including self-occlusion of, e.g., the subject's arm against the torso. Our approach begins with existing methods for segmenting a person from an image, 2D skeleton estimation, and fitting a (semi-nude) morphable, posable 3D model. The result of this first stage, while animatable, does not conform to the silhouette and does not look natural.

Our key technical contribution, then, is a method for constructing an animatable 3D model that matches the silhouette in a single photo and handles self-occlusion. Rather than deforming the 3D mesh from the first stage – a difficult problem for intricate regions such as fingers and for scenarios like abstract artwork – we map the problem to 2D, perform a silhouette-aligning warp in image space, and then lift the result back into 3D. This 2D warping approach works well for handling complex silhouettes. Further, by introducing label maps that delineate the boundaries between body parts, we extend our method to handle certain self-occlusions.

Our operating range on input and output is as follows. The person should be shown in a whole (full body photo) as a fairly frontal view. We support partial occlusion, specifically of arms in front of the body. While we aim for a mesh that is sufficient for convincing animation, we do not guarantee a metrically correct 3D mesh, due to the inherent ambiguity in reconstructing a 3D model from 2D input. Finally, as existing methods for automatic detection, segmentation, and skeleton fitting are not yet fully reliable (especially for abstract artwork), and hallucinating the appearance of the back of a person is an open research problem, we provide a user interface so that a small amount of input can correct errors and guide texturing when needed or desired.

---

[1] Artistic cinemagraphs: http://cinemagraphs.com/

To the best of our knowledge, our system is the first to enable 3D animation of a clothed subject from a single image. The closest related work either does not fully recover 3D models [61] or is built on monocular video input [3]. We compare to these prior approaches, and finally show results for a wide variety of examples such as 3D animations and AR experiences.

3.2 Related Work

Animation from photos or videos General animation from video has led to many creative effects over the years. The seminal "Video Textures" [143] work shows how to create a video of infinite length starting from a single video. Human-specific video textures were produced from motion capture videos via motion graphs [40]. [186] explore multi-view captures for human motion animation, and [197] demonstrate that clothing can be deformed in user videos guided by body skeleton and videos of models wearing the same clothing. Cinemagraphs [166, 8] or Cliplets [68] create a still with small motion in some part of the still, by segmenting part of a given video in time and space.

Animation from big data Relevant also are animations created from big data sets of images, e.g., personal photo collections of a person where the animation shows a transformation of a face through years [75], or Internet photos to animate transformation of a location in the world through years [101], e.g., how flowers grow on Lombard street in San Francisco, or the change of glaciers over a decade.

Animation from single photographs Animating from a single photo, rather than videos or photo collections, also resulted in fascinating effects. [24] animate segmented regions to create an effect of water ripples or swaying flowers. [189] predict motion cycles of animals from a still photo of a group of animals, e.g., a group of birds where each bird has a different wing pose. [76] show that it's possible to modify the 3D viewpoint of an object in a still by matching to a database of 3D shapes, e.g., rotating a car on in a street photo. [7] showed how to use a video of an actor making facial expressions and moving their head to create a similar motion in a still photo. Specific to

body shapes, [196] showed that it's possible to change the body weight and height from a single image and in a full video [65]. [61] presented a user-intensive, as-rigid-as-possible 2D animation of a human character in a photo, while ours is 3D.

Human body estimation from single photos For 3D body shape estimation from single photo, [14] provided the SMPL model which captures diverse body shapes and proved highly useful for 3D pose and shape estimation applications. Further, using deep networks and the SMPL model, [170, 70, 124, 117] present end-to-end frameworks for single view body pose and shape estimation. [169] directly infer a volumetric body shape. [47] finds dense correspondence between human subjects and UV texture maps. For multi-view, [96, 34] reconstruct a 3D mesh from sketches or silhouettes. [3] applied SMPL model fitting to a video of a subject rotating in front of a static camera, and is further extended in [2] to improve mesh and texture quality with shape from shading. Recently, the idea of parametric model has further been extended to animals [199, 71].

Most single-image person animation has focused on primarily 2D or pseudo-3D animation (e.g., [61]) while we aim to provide a fully 3D experience. Most methods for 3D body shape estimation focus on semi-nude body reconstruction and not necessarily ready for animation, while we take cloth into account and look for an animatable solution. The most similar 3D reconstruction work is [3] although they take a video as input. We compare our results to [61] and [3] in Sec. 3.5.

3.3 Overview

Given a single photo, we propose to animate the human subject in the photo. The overall system works as follows (Fig. 3.2): We first apply state-of-the-art algorithms to perform person detection, segmentation, and 2D pose estimation. From the results, we devise a method to construct a rigged mesh (Section 3.4). Any 3D motion sequence can then be used to animate the rigged mesh.

To be more specific, we use Mask R-CNN [52] for person detection and segmentation (implementation by [104]). 2D body pose is estimated using [175], and person segmentation is refined using Dense CRF [81]. Once the person is segmented out of the photo, we apply PatchMatch [10] to fill in the regions where the person used to be.

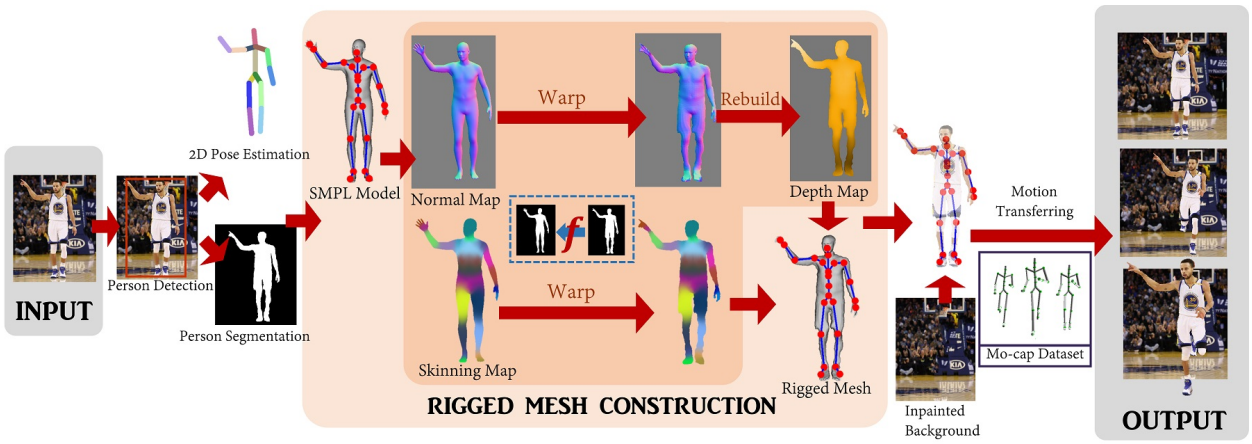


Figure 3.2: Overview of our method. Given a photo, person detection, 2D pose estimation, and person segmentation, is performed using off-the-shelf algorithms. Then, A SMPL template model is fit to the 2D pose and projected into the image as a normal map and a skinning map. The core of our system is: find a mapping between person's silhouette and the SMPL silhouette, warp the SMPL normal/skinning maps to the output, and build a depth map by integrating the warped normal map. This process is repeated to simulate the model's back view and combine depth and skinning maps to create a complete, rigged 3D mesh. The mesh is further textured, and animated using motion capture sequences on an inpainted background. *Photo credits to Getty Images*

3.4 Mesh Construction and Rigging

The key technical idea of this work is how to recover an animatable, textured 3D mesh from a single photo to fit the proposed application.

We begin by fitting the SMPL morphable body model [95] to a photo, including the follow-on method for fitting a shape in 3D to the 2D skeleton [14]. The recovered SMPL model provides an excellent starting point, but it is semi-nude, does not conform to the underlying body shape of the person and, importantly, does not match the clothed silhouette of the person.

One way is to force the SMPL model to fit the silhouettes by optimizing vertex locations on the SMPL mesh, taking care to respect silhouette boundaries, avoid pinching, and self-intersection. This is challenging especially around intricate regions such as fingers. This was indeed explored by [3], and we compare to those results in the experiments.

Instead, we take a 2D approach: warp the SMPL silhouette to match the person silhouette in the original image and then apply that warp to projected SMPL normal maps and skinning maps. The resulting normal and skinning maps can be constructed for both front and (imputed) back views and then lifted into 3D, along with the fitted 3D skeleton, to recover a rigged body mesh that exactly agrees with the silhouettes, ready for animation. The center box in Figure 3.2 illustrates our approach.

In the following, we describe how we construct a rigged mesh using 2D warping (Section 3.4.1), then present how to handle arm-over-body self-occlusion (Section 3.4.2).

3.4.1 Mesh Warping, Rigging, & Skinning

In this section, we describe the process for constructing a rigged mesh for a subject without self-occlusion.

We start with the 2D pose of the person and the person's silhouette mask S. For simplicity, we refer to S both as a set and as a function, i.e., as the set of all pixels within the silhouette, and as a binary function $S(x) = 1$ for pixel x inside the silhouette or $S(x) = 0$ for x outside the silhouette.

To construct a 3D mesh with skeletal rigging, we first fit a SMPL model to the 2D input pose

using the method proposed by [14], which additionally recovers camera parameters. We then project this mesh into the camera view to form a silhouette mask S_{SMPL}. The projection additionally gives us a depth map $Z_{\text{SMPL}}(x)$, a normal map $N_{\text{SMPL}}(x)$ and a skinning map $W_{\text{SMPL}}(x)$ for pixels $x \in S_{\text{SMPL}}$. The skinning map is derived from the per-vertex skinning weights in the SMPL model and is thus vector-valued at each pixel (one skinning weight per bone).

Guided by S_{SMPL} and the input photo's silhouette mask S, we then warp Z_{SMPL}, N_{SMPL}, and W_{SMPL} to construct an output depth map (at the silhouette only) $Z_{\partial S}(x \in \partial S)$, normal map $N(x)$, and skinning map $W(x)$, respectively, for pixels $x \in S$. $N(x)$ is then integrated to recover the final depth map $Z(x)$, subject to matching $Z_{\partial S}(x)$ at the silhouette boundary ∂S. More concretely, we solve for a smooth inverse warp, $f(x)$, such that:

$$S(x) = S_{\text{SMPL}}(f(x)) \tag{3.1}$$

and then apply this warp to the depth and skinning maps:

$$Z_{\partial S}(x \in \partial S) = Z_{\text{SMPL}}(f(x)) \tag{3.2}$$

$$N(x) = N_{\text{SMPL}}(f(x)) \tag{3.3}$$

$$Z(x) = \text{Integrate}[N; Z_{\partial S}] \tag{3.4}$$

$$W(x) = W_{\text{SMPL}}(f(x)) \tag{3.5}$$

We experimented with setting $Z(x) = Z_{\text{SMPL}}(f(x))$, but the resulting meshes were usually too flat in the z direction (See Fig. 3.3b). The warping procedure typically stretches the geometry in the plane (the SMPL model is usually thinner than the clothed subject, often thinner than even the unclothed subject), without similarly stretching (typically inflating) the depth. We address this problem by instead warping the *normals* to arrive at $N(x)$ and then integrating them to produce $Z(x)$. In particular, following [12], we solve a sparse linear system to produce a $Z(x)$ that agrees closely with the warped normals $N(x)$ subject to the boundary constraint that $Z(x) = Z_{\partial S}(x)$ for pixels $x \in \partial S$. Fig. 3.3 shows the difference between the two methods we experimented with.

(a) Reference SMPL (b) Warping depth (c) Warping normal

Figure 3.3: Comparison of different depth map constructions, after stitching front and back depth maps together (Section 3.4.1). Given (a) a reference SMPL model, we can reconstruct a mesh (b) by warping the SMPL depth maps or (c) by warping the SMPL normal maps and then integrating. Notice the flattening evident in (b), particularly around the head.

To construct the inverse warp, $f(x)$, many smooth warping functions are possible; we choose one based on mean-value coordinates [41] because it is well defined over the entire plane for arbitrary planar polygons without self-intersections, which fits our cases very well. In particular, given the ordered set of points (vertices) on the closed polygonal boundary of the input silhouette, $p_i \in \partial S = (p_0, p_1, \ldots, p_{m-1})$, we can represent any point inside of S as:

$$x = \sum_{i=0}^{m-1} \lambda_i(x) p_i \tag{3.6}$$

where $(\lambda_0(x), \lambda_1(x), \ldots, \lambda_{m-1}(x))$ are the mean-value coordinates of any $x \in S$ with respect to the boundary vertices p_i.

Suppose we have a correspondence function ϕ that identifies p_i on the input silhouette boundary ∂S with points on the SMPL silhouette boundary $p_i^{\text{SMPL}} \in \partial S_{\text{SMPL}} = (p_0^{\text{SMPL}}, p_1^{\text{SMPL}}, \ldots, p_{n-1}^{\text{SMPL}})$:

$$p_i \rightarrow p_{\phi[i]}^{\text{SMPL}}. \tag{3.7}$$

Then, using the same mean-value coordinates from Eq. 3.6, we define the warp function to be:

$$f(x) = \sum_{i=0}^{m-1} \lambda_i(x) p_{\phi[i]}^{\text{SMPL}}. \tag{3.8}$$

Next, we describe how we compute the correspondence function ϕ, fill holes in the normal and skinning maps, and then construct a complete mesh with texture.

Boundary matching We now seek a mapping ϕ that provides correspondence between points $p_i \in \partial S$ and points $p_j^{\text{SMPL}} \in \partial S_{\text{SMPL}}$. We would like each point p_i to be close to its corresponding point $p_{\phi[i]}^{\text{SMPL}}$, and, to encourage smoothness, we would like the mapping to be monotonic without large jumps in the indexing. To this end, we solve for $\phi[i]$ to satisfy:

$$\underset{\phi[0],\ldots,\phi[m-1]}{\arg\min} \sum_{i=0}^{m-1} D(p_i, p_{\phi[i]}^{\text{SMPL}}) + T(\phi[i], \phi[i+1]) \tag{3.9}$$

where

$$D(p_i, p_{\phi[i]}^{\text{SMPL}}) = \|p_i - p_{\phi[i]}^{\text{SMPL}}\|_2 \tag{3.10}$$

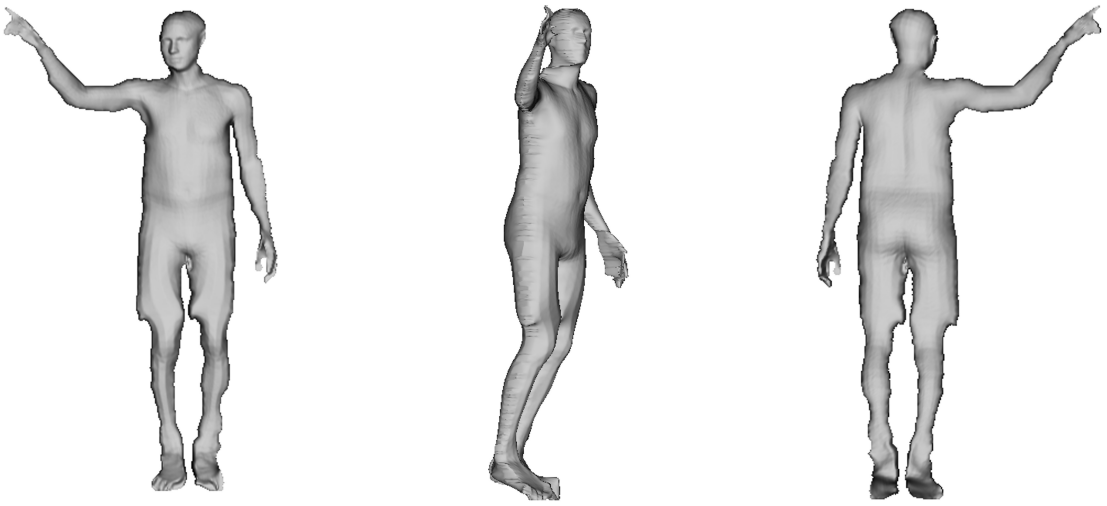
and

$$T(\phi[i], \phi[i+1]) = \begin{cases} 1, & \text{if } 0 \leq \phi[i+1] - \phi[i] \leq \kappa \\ \infty, & \text{otherwise} \end{cases} \tag{3.11}$$

$D(p_i, p_{\phi[i]}^{\text{SMPL}})$ is designed to encourage closeness of corresponding points, and $T(\phi[i], \phi[i+1])$ avoids generating an out-of-order sequence with big jumps. Because we are indexing over closed polygons, we actually use $\phi[i\%m]\%n$ in the objective, where m and n are the numbers of elements of ∂S and ∂S_{SMPL} respectively. With $\kappa = 32$, we solve for ϕ with dynamic programming.

Hole-filling In practice, holes may arise when warping by $f(x)$, i.e., small regions in which $f(x) \notin S_{\text{SMPL}}$, due to non-bijective mapping between ∂S and ∂S_{SMPL}. We smoothly fill these holes in the warped normal and skinning weight maps. Please refer to the appendix A for more detail and an illustration of the results of this step.

Constructing the complete mesh The method described so far recovers depth and skinning maps for the front of a person. To recover the back of the person, we virtually render back view of the fitted SMPL model, mirror the person mask, and then apply the warping method described previously.



(a) Front mesh (b) Combined mesh (side view) (c) Back mesh

Figure 3.4: Reconstructed mesh results. We reconstruct the front mesh (a) and the back mesh (c) separately and then combine them into one mesh, viewed from the side in (b).

We reconstruct front and back meshes in the standard way: back-project depths into 3D and construct two triangles for each 2x2 neighborhood. We assign corresponding skinning weights to each vertex. Stitching the front and back meshes together is straightforward as they correspond at the boundary. Fig. 3.4 illustrates the front and back meshes and the stitched model.

3.4.2 Self-occlusion

When the subject self-occludes – one body part over another – reconstructing a single depth map (e.g., for the front) from a binary silhouette will not be sufficient. To handle self-occlusion, we segment the body into parts via body label map, complete the partially occluded segments, and then reconstruct each part using the method described in Section 3.4.1. Fig. 3.5 illustrates our approach.

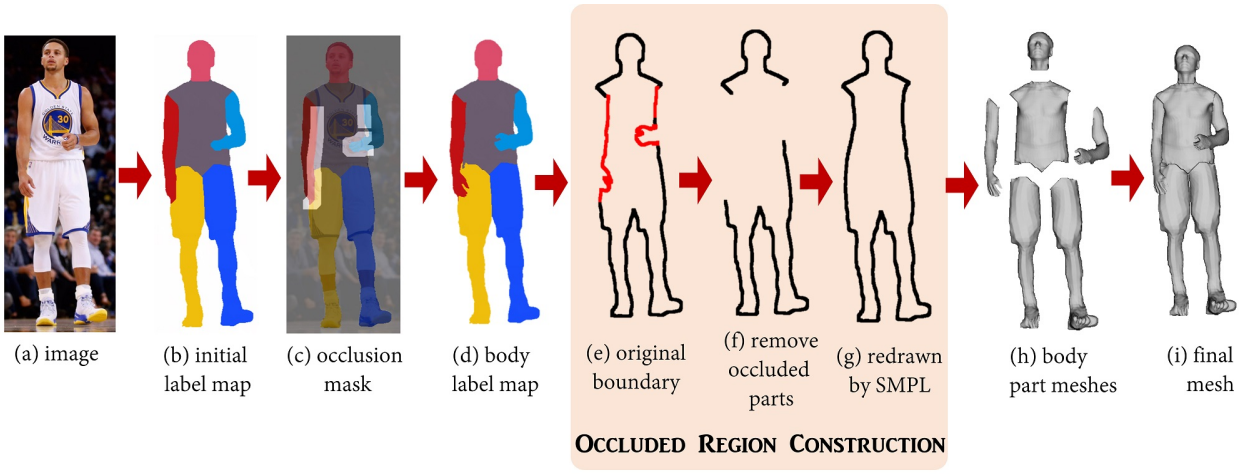


Figure 3.5: Starting from the input image (a) and its corresponding silhouette and projected SMPL body part model, we recover an initial body part label map (b). After identifying points at occlusion boundaries, we construction an occlusion mask (lighter areas in (c)) and then refine it to construct the final body label map (d). The body part regions near occlusions have spurious boundaries, shown in red in (e). We remove these spurious boundaries (f) and replace them with transformed versions of the SMPL boundaries (g). We then rebuild the body part-by-part (h) and assemble into the final mesh (i). *Photo Credits to Getty Images*

.

We focus on self-occlusion when the arms partially cross other body parts such that the covered parts are each still a single connected component. Our method does not handle all self-occlusion scenarios, but does significantly extend the operating range and show a path toward handling more cases.

Body label map The projected SMPL model provides a reference body label map L_{SMPL} that does not conform closely to the image. We use this label map to construct a final label map L in two stages: (1) estimate an initial label map L_{init} for each pixel $x \in S$ to be as similar as possible to L_{SMPL}, then (2) refine L_{init} at occlusion boundaries where the label discontinuities should coincide with edges in the input image.

Initial Body Labeling We solve for the initial (rough) body label map L_{init} by minimizing a Markov Random Field (MRF) objective:

$$\underset{L_{\text{init}}}{\text{minimize}} \quad \sum_{p \in S} U(L_{\text{init}}(p)) + \gamma \sum_{p \in S, q \in \mathcal{N}(p) \cap S} V(L_{\text{init}}(p), L_{\text{init}}(q)) \tag{3.12}$$

where

$$U(L_{\text{init}}(p)) = \underset{r | L_{\text{SMPL}}(r) = L(p)}{\text{minimize}} \|p - r\|_2 \tag{3.13}$$

$$V(L_{\text{init}}(p), L_{\text{init}}(q)) = \begin{cases} 1 & \text{if } L_{\text{init}}(p) \neq L_{\text{init}}(q) \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

$\mathcal{N}(p)$ is the 8-neighborhood of p. $U(.)$ scores a label according to the distance to the nearest point in L_{SMPL} with the same label, thus encouraging L_{init} to be similar in shape to L_{SMPL}, while $V(.)$ encourages spatially coherent labels.

We use α-expansion [16] to approximately solve for L_{init}, with $\gamma = 16$. Fig. 3.5(b) illustrates the initial label map produced by this step.

Refined Body Labeling Next, we refine the body label map to more cleanly separate occlusion boundaries.

Occlusion boundaries occur when two pixels with different part labels are neighbors in the image, but are not neighbors on the 3D body surface. To identify these pixels, we first compute warp functions f_ℓ that map each body part $L_{\text{init}} = \ell$ to the corresponding body part $L_{\text{SMPL}} = \ell$, using the mean-value coordinate approach described in Section 3.4.1, performed part-by-part. Then, along the boundaries of arm parts of L_{init}, for each pair of neighboring pixels (p, q) with different labels, we determine the corresponding projected SMPL locations $(f_{L_{\text{init}}(p)}(p), f_{L_{\text{init}}(q)}(q))$, back-project them onto the SMPL mesh, and check if they are near each other on the surface. If not, these pixels are identified as occlusion pixels. Finally, we dilate around these occlusion pixels to generate an occlusion mask O. The result is shown in Fig. 3.5(c).

We now refine the labels within O to better follow color discontinuities in the image I, giving

us the final body label map L. For this, we define another MRF:

$$\underset{L}{\text{minimize}} \quad \sum_{p \in O} U(L(p)) + \gamma \sum_{p \in O, q \in \mathcal{N}(p)} V(L(p), L(q)) \tag{3.15}$$

where

$$U(L(p)) = -\log(\text{GMM}(L(p), I(p))) \tag{3.16}$$

$$V(L(p), L(q)) = C(L(p), L(q)) e^{-\beta \|I(p) - I(q)\|^2} \tag{3.17}$$

$$C(L(p), L(q)) = \begin{cases} 1/\|p - q\| & \text{if } L(p) \neq L(q) \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

where $\text{GMM}(L(p), I(p))$ is the probability of p with color $I(p)$ labeled as $L(p)$, modeled using a Gaussian Mixture Model. We set $\gamma = 8$, and, following [134], we set β to be:

$$\beta = (2\langle \|(I(p) - I(q)\|^2 \rangle)^{-1} \tag{3.19}$$

where $\langle . \rangle$ averages over all pairs of neighboring pixels in O.

The problem is solved by iteratively applying α-expansions [16], where in each iteration we re-estimate $\text{GMM}(.)$ using the latest approximated L initizlied as L_{init}. Fig 3.5(d) illustrates the final body map.

Occluded region construction We now have the shapes of the unoccluded segments; the next challenge is to recover the shapes of the partially occluded parts.

We first combine the labels of the head, torso, and legs together into one region B. Then we extract the boundary ∂B and identify the occlusion boundaries, $\partial B^{ocl} \in \partial B \cap O$ (shown in red in Fig. 3.5(e)). Next, for a contiguous set of points $\partial B_i^{ocl} \in \partial B^{ocl}$ (e.g., one of the three separate red curves in Fig. 3.5(e)), we find the corresponding boundary $\partial B_{\text{SMPL}}^{ocl} \in \partial B_{\text{SMPL}}$ using the boundary matching algorithm from Section 3.4.1, where B_{SMPL} is the region formed by projecting the SMPL head, torso, and legs to the image plane. We then replace ∂B_i^{ocl} with $\partial B_{\text{SMPL}}^{ocl}$ by a similarity transform defined by the end points of ∂B_i^{ocl} and $\partial B_{\text{SMPL}}^{ocl}$, as shown in Fig.3.5-(f) and (g).

Mesh construction Once we have completed body labeling and recovered occluded shapes, we project the SMPL model part-by-part to get per-part SMPL depth, normal, and skinning weight maps, then follow the approach in Section 3.4.1 to build part meshes (Fig.3.5-(h)), and assemble them together to get our final body mesh (Fig.3.5-(i)). Finally, we apply Laplacian smoothing to reduce jagged artifacts along the mesh boundaries due to binary silhouette segmentation.

3.4.3 Final Steps

Head pose correction Accuracy in head pose is important for good animation, while the SMPL head pose is often incorrect. Thus, as in [77, 74], we detect facial fiducials in the image and solve for the 3D head pose that best aligns the corresponding, projected 3D fiducials with the detected ones. After reconstructing the depth map for the head as before, we apply a smooth warp that exactly aligns the projected 3D fiducials to the image fiducials. Whenever the face or fiducials are not detected, this step is skipped.

Texturing For the front of the subject, we project the image onto the geometry. Occluded, frontal body part regions are filled using PatchMatch [10]. Hallucinating the back texture is an open research problem [97, 38, 98]. We provide two options: (1) paste a mirrored copy of the front texture onto the back, (2) inpaint with optional user guidance. For the second option, inpainting of the back is guided by the body label maps, drawing texture from regions with the same body labels. The user can easily alter these label maps to, e.g., encourage filling in the back of the head with hair texture rather than face texture. Finally the front and back textures are stitched with poisson blending [127].

Please refer to the appendix A for more details and illustrations of head pose correction and texturing.

3.5 Results

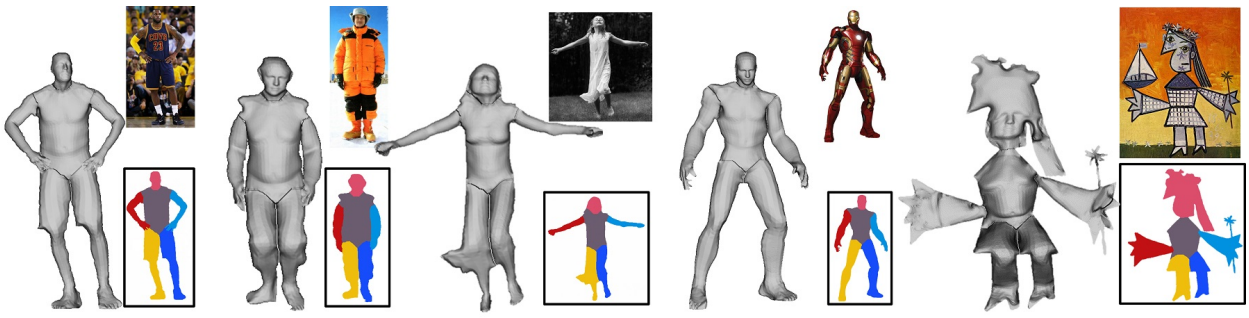


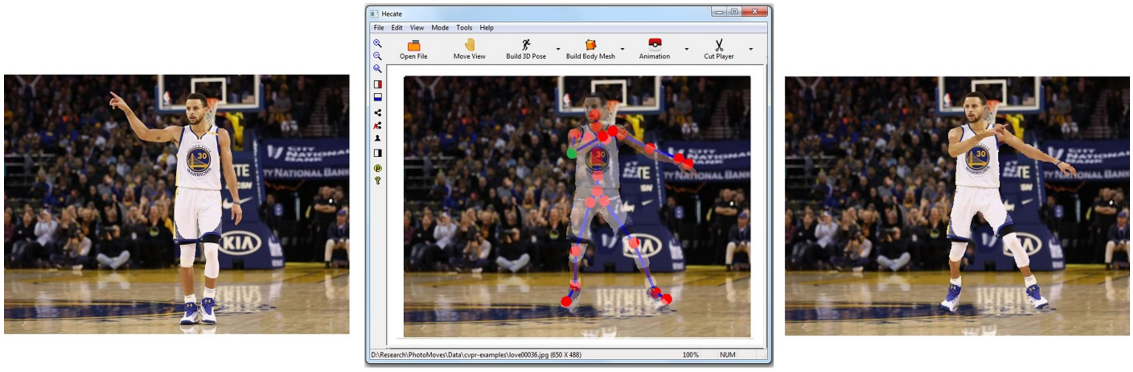
Figure 3.6: Examples of body label maps and meshes (input photos are put on top right corner). *Photo credits to [45, 27, 128, 157, 37]*

Below we describe our user interface, results, and comparisons to related methods. We have tested our method on 70 photos downloaded from the Internet (spanning art, posters, and graffiti that satisfied our photo specifications–full body, mostly frontal). Figs. 3.11 and 3.10 show our typical animation and augmented reality results. With our UI, the user can change the viewpoint during animation, and edit the human pose. With an AR headset, the user can place the artwork on the wall and walk around the animation while it is playing. **Please refer to the supplementary video[2] for dynamic versions of the results.**

User interface We have created a user interface where the user can interactively: (1) Modify the animation: the default animation is "running", where the user can keep some body parts fixed, change the sequence (e.g., choose any sequence from [25]), modify pose and have the model perform an action starting from the modified pose. (2) Improve the automatic detection box, skeleton, segmentation, and body label map if they wish. (3) Choose to use mirrored textures for the back or make adjustments via editing of the body label map. The user interaction time for (2) and (3) is seconds, when needed.

[2] https://youtu.be/G63goXc5MyU

Fig. 3.7 shows an example of the pose editing process. In our UI the mesh becomes transparent to reveal the body skeleton. By selecting and dragging the joints the user can change the orientation of the corresponding bones. A new image where the pose is edited can be then easily generated.



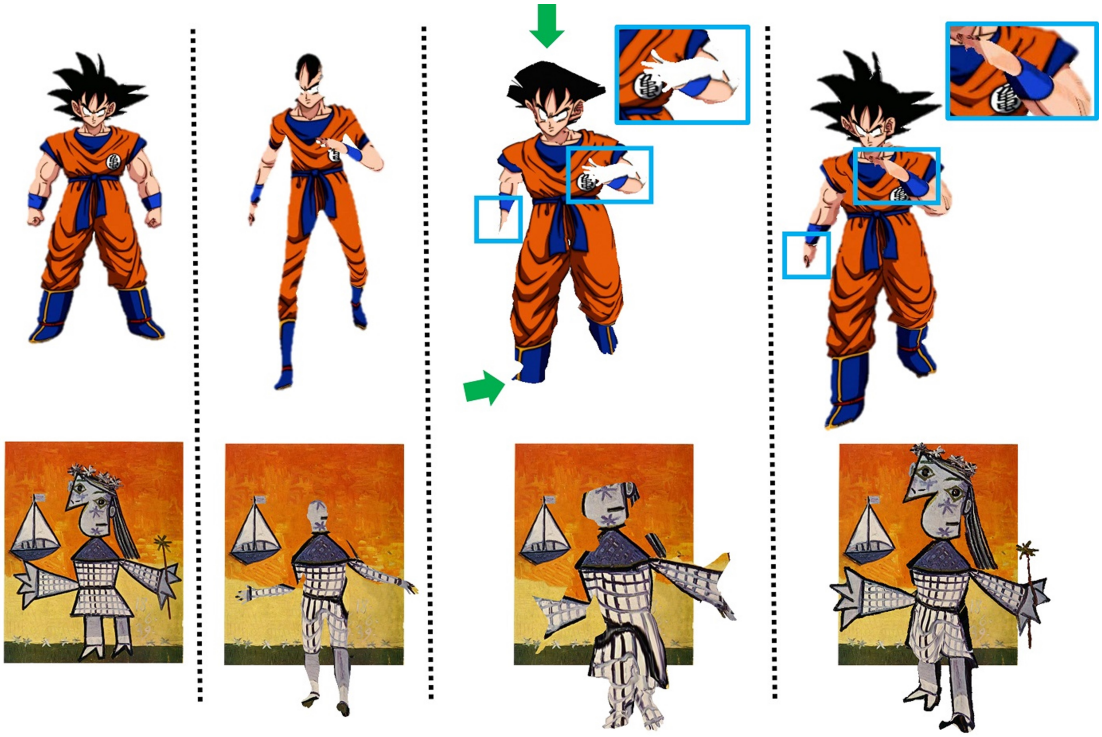
(a) Input image (b) User interface to drag joints (c) Results

Figure 3.7: Our pose editing UI: (a) Input photo. (b) Editing pose by dragging joints. (c) Result. *Photo credits to Getty Images*

The underlying reconstructed geometry for several examples is shown in Fig. 3.6. The resulting meshes do not necessarily represent the exact 3D geometry of the underlying subject, but they are sufficient for animation in this application and outperform state of the art as shown below.

Comparison with Bogo et al. [14]: As shown in Fig. 3.8(b), the fitted, semi-nude SMPL model [14] does not correctly handle subject silhouettes.

Comparison with Alldieck et al. [3]: In [3], a SMPL mesh is optimized to approximately match silhouettes of a rotating human subject in a monocular video sequence. Their posted code uses 120 input frames, with objective weights tuned accordingly; we thus provide their method with 120 copies of the input image, in addition to the same 2D person pose and segmentation we use. The results are shown in Fig. 3.8(c). Their method does not fit the silhouette well; e.g., smooth SMPL parts don't become complex (bald head mapped to big, jagged hair) and the detailed fingers are not warped well to the closed fists or abstract art hands. These failures are mostly due to a strong human shape prior, which is not suited to tackling cases like cartoon characters or abstract paintings.



(a) Input image (b) Bogo et al. (c) Alldieck et al. (d) Our method

Figure 3.8: Comparison with [14, 3]. (a) Input photo. (b) A fitted SMPL model [14]. (c) A deformed mesh using [3]. The mesh fails to deform hair and shoes (green arrows) and fingers (blue box). (d) Our mesh. *Photo credits to [30, 37].*

Comparison with [61]: We have run our method on the only example in [61] that demonstrated substantial out-of-plane motion rather than primarily in-plane 2D motion (see Fig. 3.9). Our result appears much less distorted in still frames (due to actual 3D modeling) and enables 3D experiences (e.g., AR) not possible in [61]. We verified our qualitative observation with a user study on MTurk, asking users to decide which animation is "more realistic." 103 participants responded, and 86% preferred ours.

(a) Input image (b) Hornung et al. (c) Our 3D demonstration

Figure 3.9: Comparison result with [61]: (a) input photo; (b) animation method proposed in [61]; (c) 3D demonstration using our method, which is not possible in [61]. *Photo credits to Hornung et al.*

Figure 3.10: AR results of our method for different environments (input photos inset). The floor (left) and couch (right) are real, while the people are augmented. *Photo credits to [159, 28]*

Figure 3.11: Six animation results. The input is always on left. *Photo credits to [45, 180, 30, 28, 37, 157]*

3.6 Discussion

Limitations We note the following limitations (see also Fig. 3.12): (1) Shadows and reflections are currently not modeled by our method and thus won't move with the animation. (2) Since the reconstructed mesh must fit the silhouette, the shape may look unrealistic, e.g., wrong shape of shoes; on the other hand this enables handling abstract art. (3) Our method accounts for self-occlusions when arms partially occlude the head, torso, or legs. It remains future work to handle other occlusions, e.g., legs crossed when sitting. (4) Person detection and segmentation, pose detection and body labeling can fail, requiring manual corrections. (5) We have opted for simple texture inpainting for occluded body parts, with some user interaction if needed. Using deep learning to synthesize, e.g., the appearance of the back of a person given the front, is a promising research area, but current methods that we have tested [97, 38, 98] give very blurry results.

(a) Unmodeled shadows (b) Distorted mesh (c) Segmentation errors

Figure 3.12: Examples of limitations (inputs in blue boxes). (a) Shadows not modeled. (b) Unrealistic mesh. (c) manually correcting segmentation errors. *Photo credits to [180, 28, 128]*

Summary We have presented a method to create a 3D animation of a person in a single image. Our method works with large variety of of whole-body, fairly frontal photos, ranging from sports

photos, to art, and posters. In addition, the user is given the ability to edit the human in the image, view the reconstruction in 3D, and explore it in AR.

The proposed application enables new ways for people to enjoy and interact with photos. Moreover, it suggests a pathway to reconstructing a virtual avatar from a single image while providing insight into the state of the art of human modeling from a single photo.

Chapter 4

HUMANNERF: FREE-VIEWPOINT RENDERING OF MOVING PEOPLE FROM MONOCULAR VIDEO

(a) monocular video (b) free-viewpoint rendering

Figure 4.1: Our method takes as input a monocular video of a human performing complex movement, e.g., dancing (left), and creates a free-viewpoint rendering for any frame in the sequence (right). We construct a canonical subject appearance volume, and a motion field mapping from observation to canonical space, trained on the video. At test time, we take just the pose from the source frame (red square) and synthesize all output views, including the target view.

This chapter presents a joint work with Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. The work was published in CVPR 2022 [178]. The discussion of and comparisons to related work are based on the state-of-the-art at the time.

4.1 Introduction

Given a single video of a human performing an activity, e.g., a YouTube or TikTok video of a dancer, we would like the ability to pause at any frame and rotate 360 degrees around the performer

to view them from any angle at that moment in time (Figure 1). This problem – free-viwepoint rendering of a moving subject – is a longstanding research challenge, as it involves synthesizing previously unseen camera views while accounting for cloth folds, hair movement, and complex body poses [69, 105, 18, 154, 171, 48, 19, 40]. The problem is particularly hard for the case of "in-the-wild" videos taken with a single camera (monocular video), the case we address in this chapter.

Previous neural rendering methods [102, 93, 126, 192, 103, 172, 11] typically assume multi-view input, careful lab capture, or do not perform well on humans due to non-rigid body motion. Human-specific methods typically assume a SMPL template [95] as a prior, which helps constrain the motion space but also introduces artifacts in clothing and complex motions that are not captured by the SMPL model [126, 125]. Recently deformable NeRF methods [121, 122, 168, 130] perform well for small deformations, but not for large, full body motions like dancing.

We introduce a method, called HumanNeRF, that takes as input a single video of a moving person and, after per-frame, off-the-shelf segmentation (with some manual clean-up) and automatic 3D pose estimation, optimizes for a canonical, volumetric T-pose of the human together with motion field that maps the estimated canonical volume to each video frame via a backward warping. The motion field combines skeletal rigid motion with non-rigid motion, each represented volumetrically. Our solution is data-driven, with the canonical volume and motion fields derived from the video itself and optimized for large body deformations, trained end-to-end, including 3D pose refinement, without template models. At test time, we can pause at any frame in the video and, conditioned on the pose in that frame, render the resulting volumetric representation from any viewpoint.

We show results on a variety of examples: existing lab datasets, videos we captured outside the lab, and downloads from YouTube (with creator permission). Our method outperforms the state-of-the-art numerically and produces significantly higher visual quality. **Please refer to the project page[1] to see the results in motion.**

---

[1] https://grail.cs.washington.edu/projects/humannerf/

4.2 Related Work

The physics of free-viewpoint rendering involves modeling geometry and surface properties and then rendering from new camera views. However, it remains difficult to recreate complex geometry and subtle lighting effects. Alternatively, image-based rendering [149, 158] offers to render novel views based on given set of views in the image domain with a large corpus of research over the last couple decades [84, 46, 32, 55, 54, 21, 22, 198].

Human specific rendering The work of Kanade et al. [69] is one of the earliest investigations into free-viewpoint rendering of humans. It introduced a dome equipped with cameras to recover depth maps and meshes, enabling novel views to be rendered by reprojecting and blending different views to account for mesh holes due to occlusions. Later, Matusik et al. [105] reconstructed a *visual hull* from silhouettes of the subject and rendered it by carefully selecting pixels without an auxiliary geometric representation. Carranza et al. [18] used a parameterized body model as a prior and combined marker-less motion capture and view-dependent texturing [32]. Follow-on work introduced non-rigid deformation [171], texture warping [186, 19], and various representations based on volumes [31] or spheres [154]. Collet et al. [26] and Guo et al. [48] build a system as well as pipeline that produces high-quality streamable [26] or even relightable [48] free-viewpoint videos of moving people.

Most of these methods rely on multi-view videos – typically expensive studio setups – while we are interested in a simple monocular camera configuration.

Neural radiance fields NeRF [110] and its extensions [192, 11, 194, 115, 56, 152, 161] enable high quality rendering of novel views of static scenes. NeRF has recently been extended to dynamic scenes [121, 122, 168, 44, 88, 130, 183], though these approaches generally assume that motion is small. We compare our method to these dynamic and deformable NeRF works in our results section.

Human-specific neural rendering The work of Liu et al. [93] starts from a pre-captured body model and learns to model time-dependent dynamic textures and enforce temporal coherence. Martin-Brualla et al. [102] trained a UNet to improve the artifacts introduced by volumetric capture. The follow-up work of Pandey et al. [119] reduced the number of required input frames to as few as a single RGBD image via semi-parametric learning. Wu et al. [182] and Peng et al. [126] explored the use of learned structured latent codes embedded for point clouds (from MVS [145]) or reposed mesh vertices (from SMPL [95]) and learn an accompanying UNet- or NeRF-based neural renderer. Zhang et al. [66] decomposed a scene into background and individual performers, and represented them with separated NeRFs thus enabling scene editing. Other than free-viewpoint rendering, there is another related active research field that focus on human motion retargeting either in 2D [173, 20, 140, 9, 112, 97, 174] or 3D [177, 92, 125, 49, 139, 190, 62, 53]. In addition, a few concurrent works attempt to solve the problem using monocular videos. Xu et al. [187] co-learn implicit geometry as well as appearance from images. They largely focus on multi-view setups with a few examples on monocular videos where the human motion is simple (A-pose). Su et al. [155] use an over-parameterized NeRF to rigidly transform NeRF features for refining body pose and thus final rendering. The non-rigid motion is not explicitly modeled and the rendering quality is not high. A similar approach is discovered by Noguchi et al. [116] as well but still shows results of limited visual quality.

The main difference between our method and those works is that we take as input *monocular* video that contains *complex* human motions and enable high-fidelity full 3D rendering. Moreover, our formulation of skeletal motion draws inspiration from Vid2Actor proposed by Weng et al. [177], a method intended for rigidly animatable characters. Instead, we focus on the free-viewpoint application and recovering pose-dependent, non-rigid deformation and outperform them significantly for this application.

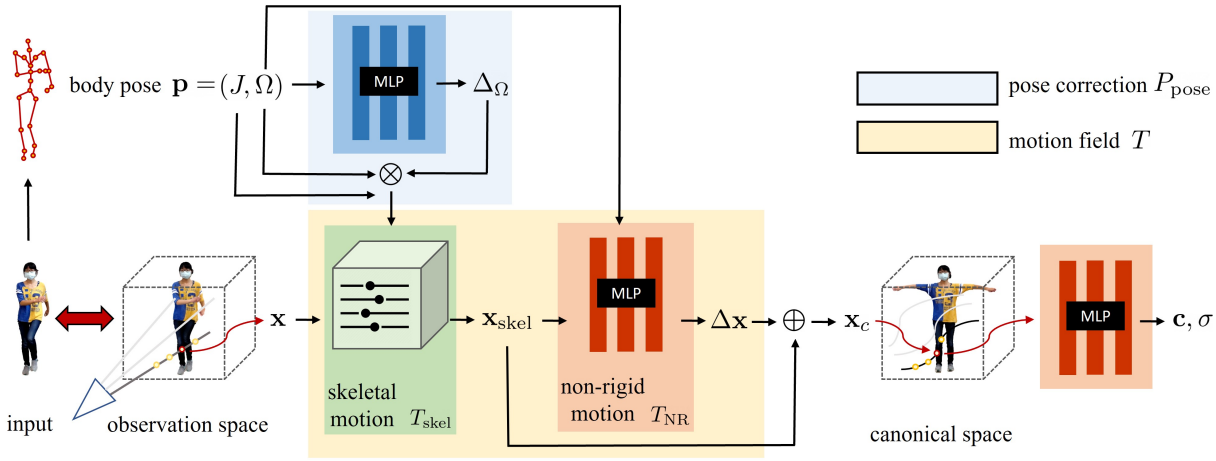


Figure 4.2: Our method takes a video frame as input and optimizes for canonical appearance, represented as a continuous field, as well as a motion field mapping from observation to canonical space. The motion field is decomposed into skeletal rigid and non-rigid motion, represented as a discrete grid and a continuous field respectively. We additionally refine body pose initialized with an off-the-shelf body pose estimator, leading to better alignment. A loss is imposed between the volume rendering in observation space and the input image, directing optimization towards a solution.

4.3 Representing a Human as a Neural Field

We represent a moving person with a canonical appearance volume F_c warped to an observed pose to produce output appearance volume F_o:

$$F_o(\mathbf{x}, \mathbf{p}) = F_c(T(\mathbf{x}, \mathbf{p})), \tag{4.1}$$

where $F_c : \mathbf{x} \rightarrow (\mathbf{c}, \sigma)$ maps position $\mathbf{x}$ to color $\mathbf{c}$ and density σ, and $T : (\mathbf{x}_o, \mathbf{p}) \rightarrow \mathbf{x}_c$ defines a motion field mapping points from observed space back to canonical space, guided by observed pose $\mathbf{p} = (J, \Omega)$, where J includes K standard 3D joint locations, and $\Omega = \{\boldsymbol{\omega}_i\}$ are local joint rotations represented as axis-angle vectors $\boldsymbol{\omega}_i$.

We handle complex human movement with complex deformation by decomposing the motion

field into two parts:

$$T(\mathbf{x}, \mathbf{p}) = T_{\text{skel}}(\mathbf{x}, \mathbf{p}) + T_{\text{NR}}(T_{\text{skel}}(\mathbf{x}, \mathbf{p}), \mathbf{p}), \tag{4.2}$$

where T_{skel} represents skeleton-driven deformation, essentially inverse (volumetric) linear-blend skinning, and T_{NR} starts from the skeleton-driven deformation and produces an offset $\Delta\mathbf{x}$ to it. In effect, T_{skel} provides the coarse deformation driven by standard skinning, and T_{NR} provides the more non-rigid effects, e.g., due to deformation of clothing.

For "in-the-wild" imagery, we use an off-the-shelf 3D body+camera pose estimator. Due to inaccuracy in pose estimation, we also solve for a pose correction function $P_{\text{pose}}(\mathbf{p})$ that better explains the observations, and apply this improvement to the skeleton-driven deformation, i.e., we replace $T_{\text{skel}}(\mathbf{x}, \mathbf{p})$ with $T_{\text{skel}}(\mathbf{x}, P_{\text{pose}}(\mathbf{p}))$ in Eq. 4.2.

Figure 4.2 gives an overview of the components of our system. In the following sections, we describe these components in detail.

Canonical volume We represent the canonical volume F_c as a continuous field with an MLP that outputs color $\mathbf{c}$ and density σ given a point $\mathbf{x}$:

$$F_c(\mathbf{x}) = \text{MLP}_{\theta_c}(\gamma(\mathbf{x})), \tag{4.3}$$

where γ is a sinusoidal positional encoding defined as $(\mathbf{x}, \sin(2^0\pi\mathbf{x}), \cos(2^0\pi\mathbf{x}), ..., \sin(2^{L-1}\pi\mathbf{x}), \cos(2^{L-1}\pi\mathbf{x}))$ and L is a hyper-parameter that determines the number of frequency bands [110].

Skeletal motion Following Weng et al. [177], we compute the skeletal deformation T_{skel} as a kind of inverse, linear blend skinning that maps points in the observation space to the canonical space:

$$T_{\text{skel}}(\mathbf{x}, \mathbf{p}) = \sum_{i=1}^{K} w_o^i(\mathbf{x})(R_i\mathbf{x} + \mathbf{t}_i), \tag{4.4}$$

where w_o^i is the blend weight for the i-th bone and R_i, $\mathbf{t}_i$ are the rotation and translation, respectively, that map the bone's coordinates from observation to canonical space; R_i and $\mathbf{t}_i$ can be explicitly computed from $\mathbf{p}$ (see Appendix B). We then aim to optimize for w_o^i.

In practice, we solve for w_c^i defined in canonical space by storing K blend weights as a set of volumes $\{w_c^i(\mathbf{x})\}$, from which the observation weights are derived as:

$$w_o^i(\mathbf{x}) = \frac{w_c^i(R_i\mathbf{x} + \mathbf{t}_i)}{\sum_{k=1}^{K} w_c^k(R_k\mathbf{x} + \mathbf{t}_k)}. \tag{4.5}$$

Solving for a single set of weight volumes $\{w_c^i(\mathbf{x})\}$ in canonical space, instead of N sets of $\{w_o^i(\mathbf{x})\}$ in observation space (corresponding to N input images), can lead to better generalization as it avoids over-fitting [177, 23].

We pack the set of $\{w_c^i(\mathbf{x})\}$ into a single volume $W_c(\mathbf{x})$ with K channels. Rather than encode W_c with an MLP, we choose an explicit volume representation for two reasons: (1) Eq. 4.5 shows that K MLP evaluations would be needed to compute each $w_o^i(\mathbf{x})$, infeasible for optimization ($K = 24$ in our work); (2) an explicit volume with limited resolution resampled via trilinear interpolation provides smoothness that can help regularize the optimization later. In practice, during optimization, rather than directly solve for volume W_c, we solve for parameters θ_{skel} of a CNN that generates the volume from a random (constant) latent code $\mathbf{z}$:

$$W_c(\mathbf{x}) = \text{CNN}_{\theta_{\text{skel}}}(\mathbf{x}; \mathbf{z}). \tag{4.6}$$

We also add one more channel, a background class, and represent W_c as a volume with $K + 1$ channels. We then apply channel-wise *softmax* to the output of the CNN, enforcing a partition of unity across the channels. The denominator of Eq. 4.5 can then be used to approximate likelihood $f(\mathbf{x})$ of being part of the subject, where $f(\mathbf{x}) = \sum_{k=1}^{K} w_c^k(R_k\mathbf{x} + \mathbf{t}_k)$. When $f(\mathbf{x})$ is close to zero, we are likely in free space away from the subject, which we will use during volume rendering.

The idea of optimizing blend weights (or skinning field) is not new. Similar approaches have been applied to human modeling [33, 62, 23, 109, 138, 190, 125, 165, 13]. Our formulation follows Weng et al. [177], but also shares similarities with Tiwari et al. [165]; the latter learns from 3D scans while we learn from 2D images, like the former.

Non-rigid motion We represent non-rigid motion T_{NR} as an offset $\Delta\mathbf{x}$ to the skeleton-driven motion, conditioned on that motion, i.e., $\Delta\mathbf{x}(\mathbf{x}, \mathbf{p}) = T_{\text{NR}}(T_{\text{skel}}(\mathbf{x}, \mathbf{p}), \mathbf{p}))$. To capture detail, we

represent T_{NR} with an MLP:

$$T_{\text{NR}}(\mathbf{x}, \mathbf{p}) = \text{MLP}_{\theta_{\text{NR}}}(\gamma(\mathbf{x}); \Omega), \tag{4.7}$$

where again we use the standard positional encoding γ and condition the MLP on Ω, the joint angles of body pose $\mathbf{p}$.

Pose correction The body pose $\mathbf{p} = (J, \Omega)$ estimated from an image is often inaccurate. To address this, we solve for an update to the pose:

$$P_{\text{pose}}(\mathbf{p}) = (J, \Delta_\Omega(\mathbf{p}) \otimes \Omega), \tag{4.8}$$

where we hold the joints J fixed and optimize for a relative update to the joint angles, $\Delta_\Omega = (\Delta\boldsymbol{\omega}_0, ..., \Delta\boldsymbol{\omega}_K)$ which is then applied to Ω to get updated rotation vectors.

Empirically we found, instead of directly optimizing for Δ_Ω, solving for the parameters θ_{pose} of an MLP that generates Δ_Ω conditioned on Ω leads to faster convergence:

$$\Delta_\Omega(\mathbf{p}) = \text{MLP}_{\theta_{\text{pose}}}(\Omega). \tag{4.9}$$

With this pose correction, we can re-write the equation that warps from observation space to canonical space as:

$$T(\mathbf{x}, \mathbf{p}) = T_{\text{skel}}(\mathbf{x}, P_{\text{pose}}(\mathbf{p})) + T_{\text{NR}}(T_{\text{skel}}(\mathbf{x}, P_{\text{pose}}(\mathbf{p})), \mathbf{p}) \tag{4.10}$$

4.4 Optimizing a HumanNeRF

In this section, we describe the overall objective function we minimize, our volume rendering procedure, how we regularize the optimization process, specific loss function details, and the ray sampling method.

HumanNeRF objective Given input frames $\{I_1, I_2, ..., I_N\}$, body poses $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\}$, and cameras $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N\}$, we are solving the problem:

$$\underset{\Theta}{\text{minimize}} \quad \sum_{i=1}^{N} \mathcal{L}\{\Gamma[F_c(T(\mathbf{x}, \mathbf{p}_i)), \mathbf{e}_i], I_i\}, \tag{4.11}$$

where $\mathcal{L}\{\cdot\}$ is the loss function and $\Gamma[\cdot]$ is a volume renderer, and we minimize the loss with respect to all network parameters $\Theta = \{\theta_c, \theta_{\text{skel}}, \theta_{\text{NR}}, \theta_{\text{pose}}\}$. As we have seen, F_c is determined by parameters θ_c, while the transformation T from observation space to canonical space relies on parameters θ_{skel}, θ_{NR}, and θ_{pose}.

4.4.1 Volume rendering

We render a neural field using the volume rendering equation [107] as described by Mildenhall et al. [110]. The expected color $C(\mathbf{r})$ of a ray $\mathbf{r}$ with D samples can be written as:

$$\begin{aligned} C(\mathbf{r}) &= \sum_{i=1}^{D}(\prod_{j=1}^{i-1}(1-\alpha_j))\alpha_i \mathbf{c}(\mathbf{x}_i), \\ \alpha_i &= 1 - \exp(-\sigma(\mathbf{x}_i)\Delta t_i), \end{aligned} \tag{4.12}$$

where Δt_i is the interval between sample i and $i+1$.

We further augment the definition of α_i to be small when approximate foreground probability $f(\mathbf{x})$ is low:

$$\alpha_i = f(\mathbf{x}_i)(1 - \exp(-\sigma(\mathbf{x}_i)\Delta t_i)), \tag{4.13}$$

We apply the stratified sampling approach proposed by NeRF [110]. We do not use hierarchical sampling since the bounding box of a subject can be estimated from their 3D body pose. We then only sample points inside the box.

4.4.2 Delayed optimization of non-rigid motion field

When solving for all the network parameters in Eq. 4.11 at once, we find that the the optimized skeleton-driven and non-rigid motions are not decoupled – a portion of the subject's skeletal motions is modeled by the non-rigid motion field – due to over-fitting of non-rigid motions to the input images. As a result, the quality degrades when rendering unseen views.

We manage the optimization process to solve the problem. Specifically, we disable non-rigid motions at the beginning of optimization, and then bring them back in a coarse-to-fine manner [121,

57]. To achieve this, for the non-rigid motion MLP, we apply a truncated Hann window to its frequency bands of positional encoding, to prevent overfitting to the data [161], increasing the window size as the optimization proceeds. Following Park et al. [121], we define the weight for each frequency band j of positional encoding:

$$w(\tau) = \frac{1 - \cos(\text{clamp}(\tau - j, 0, 1)\pi)}{2}, \tag{4.14}$$

where $\tau \in [0, L)$ determines the width of a truncated Hann window, and L is the total number of frequency bands in positional encoding. We then define τ as a function of the optimization iteration:

$$\tau(t) = L\frac{\max(0, t - T_s)}{T_e - T_s}, \tag{4.15}$$

where t is the current iteration, and T_s and T_e are hyper-parameters that determine when to enable non-rigid motion optimization and when to use full frequency bands of positional encoding. We remove position identity from positional encoding without affecting performance [11]. By doing so, we can completely disable non-rigid motion optimization by setting $\tau = 0$ [122].

4.4.3 Loss function and ray sampling method

Loss function We employ both an MSE loss to match pixel-wise appearance and a perceptual loss, LPIPS [193], to provide robustness to slight misalignments and shading variation and to improve detail in the reconstruction. Our final loss function is $\mathcal{L} = \mathcal{L}_{\text{LPIPS}} + \lambda\mathcal{L}_{\text{MSE}}$. We use $\lambda = 0.2$ and choose VGG as the backbone of LPIPS.

Patch-based ray sampling Training on random ray samples, as done in NeRF [110], does not work for minimizing our loss because LPIPS uses convolutions to extract features. Instead, we sample G patches with size $H \times H$ on an image, and render a total of $G \times H \times H$ rays in each batch. The rendered patch is compared against the patch with the same position on the input image. We use $G = 6$ and $H = 32$ in our experiments. Similar approaches were also used in NeRF-based generative models [146].

4.4.4 Implementation details

There are several small but important implementation details that contribute to best results. We describe them below.

Optimizing $\Delta\mathbf{W_c}$: Our method solves for W_c to determine skeletal rigid motion. In practice, we ask a deep network to generate ΔW_c instead, the difference between W_c and the logarithm of W_g. W_g consists of an ellipsoidal Gaussian around each body bone, given by the canonical T-pose, that specifies approximate body part regions in the canonical space. W_c is then computed as:

$$W_c = softmax(\Delta W_c + \log(W_g)), \tag{4.16}$$

where the background weight in W_g is set to one minus the sum of all the bone weights. We apply the logarithm to W_g, to compensate the exponential function in *softmax*.

Representation of global body orientation: Global subject orientation can be represented as body rotation or, equivalently, camera rotation. We choose to rotate the camera in order to keep the estimated bounding box when subject orientation changes. Specifically, we uses axis-aligned bounding boxes because for ease of implementation; however, the box will be different for the same pose but rotated global body orientation. This undesirable effect can be avoided if we instead describe changes of global body orientation as camera rotations.

Random background: During optimization, we randomly assign a solid background color to the rendering and to the input image to facilitate separation of foreground and background.

MLP initialization: We initialize the weights of the last layer of the non-rigid motion MLP and pose correction MLP to small values, $\mathcal{U}(-10^{-5}, 10^{-5})$, i.e., initializing the offset to be close to zero and the pose refinement rotation matrices each near the identity.

Importance ray sampling: We sample more rays for the foreground subject, indicated by the segmentation masks. Specifically, we enforce random ray sampling with probability 0.8 for foreground subject pixels and 0.2 for the background region.

4.5 Results

4.5.1 Evaluation dataset

We evaluate our method on the ZJU-MoCap dataset [126], self-captured data (*rugby, hoodie*), and YouTube videos downloaded from Internet (*story*[2], *way2sexy*[3], *invisible*[4]). All subjects in these videos provided consent to use their data. For ZJU-MoCap, we select 6 subjects (377, 386, 387, 392, 393, 394) with diverse motions and use images captured by "*camera 1*" as input and the other 22 cameras for evaluation. We directly apply camera matrices, body pose, and segmentation provided by the dataset. For videos "in the wild" (self-captured and YouTube videos), we run SPIN [80] to get approximate camera and body pose, automatically segment the foreground subject, and then manually correct errors in the segmentation. (High quality segmentation is necessary for best results; purely automatic segmenters were not accurate enough, and improving on them was outside the scope of this work, an area of future work.) We additionally resize video frames to keep the height of subject at approximately 500 pixels.

4.5.2 Optimization details

We optimize Eq. 4.11 using the Adam optimizer [78] with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We set the learning rate to 5×10^{-4} for θ_c (the canonical MLP), and 5×10^{-5} for all the others. We use 128 samples per ray. The optimization takes $400K$ iterations (about 72 hours) on 4 GeForce RTX 2080 Ti GPUs. We apply delayed optimization with $T_s = 10K$ and $T_e = 50K$ to ZJU-MoCap data, and with $T_s = 100K$ and $T_e = 200K$ to the others. In addition, we postpone pose refinement until after 20K iterations for in-the-wild videos.

---

[2] https://youtu.be/0ORaAnJYROg

[3] https://youtu.be/gEpJDE8ZbhU

[4] https://youtu.be/ANwEiICt7BM

4.5.3 Evaluation method

We compare our method with Neural Body [126] (typically used with multiple cameras) and HyperNeRF [122] (single moving camera around the subject), state-of-the-art methods for modeling humans and general scenes for novel view synthesis. Our method works with a single camera which can be static or moving; we focus on results with a static camera and moving subjects, a natural way to capture a person's performance. The method differences are listed in Table 4.1.

	Neural Body [126]	**HyperNeRF [122]**	**HumanNeRF**
Setup	multi-camera	single camera	single camera
Subject	dynamic human	quasi-static general scene	dynamic human
Priors	body pose, SMPL vertices (reposed)	rigidity	body pose (approx.)

Table 4.1: Differences between the compared methods.

4.5.4 Comparisons

We found HyperNeRF does not produce meaningful output for novel view synthesis in our experiments, as shown in Fig. 4.3, likely because it relies on multiple views (moving camera) to build a coherent 3D model. For the static camera case with moving subject, it fails to recover a meaningful depth map and appears to memorizes the input images rather than generalize from them. Note dynamic human motions are more extreme than the examples shown to work with HyperNeRF.

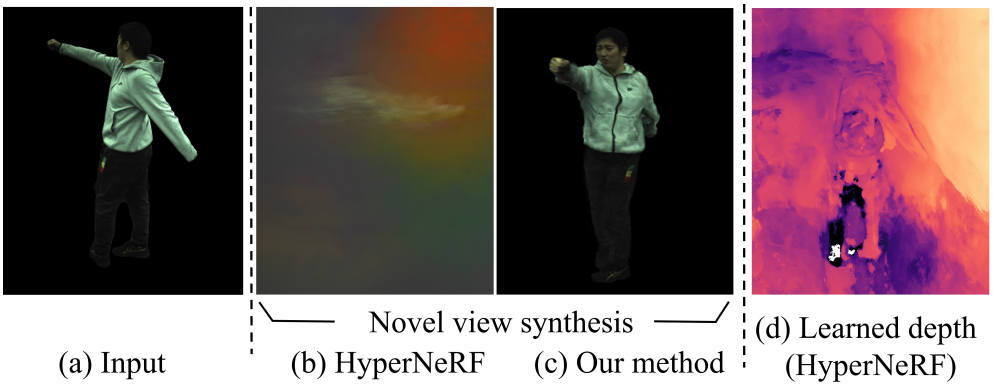


Figure 4.3: Qualitative comparison to HyperNeRF [122]

Quantitatively, We report PSNR, SSIM, and LPIPS* (LPIPS $\times 10^3$) metrics. As shown in Table 4.2, HumanNeRF outperforms Neural Body for all subjects and under all metrics, except for subject 393 on PSNR (a metric known to favor smooth results [193]). The gain is particularly significant with perceptual metric LPIPS, nearly 40% improvement on average. Fig. 4.4 shows that HumanNeRF's visual quality is substantially better then Neural Body for this dataset. Our method is capable of producing high fidelity details similar to the ground truth even on completely unobserved views, while Neural Body tends to produce blurrier results. The results for self-captured and YouTube videos, shown in Fig. 4.5, also show consistently higher quality reconstructions with HumanNeRF.

	Subject **377**			Subject **386**		
	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓
Neural Body [126]	29.11	0.9674	40.95	30.54	0.9678	46.43
Ours	30.41	0.9743	24.06	33.20	0.9752	28.99
	Subject **387**			Subject **392**		
	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓
Neural Body[126]	27.00	0.9518	59.47	30.10	0.9642	53.27
Ours	28.18	0.9632	35.58	31.04	0.9705	32.12
	Subject **393**			Subject **394**		
	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓
Neural Body[126]	28.61	0.9590	59.05	29.10	0.9593	54.55
Ours	28.31	0.9603	36.72	30.31	0.9642	32.89

Table 4.2: Quantitative comparison on ZJU-MoCap dataset. We color cells that have the best metric value.

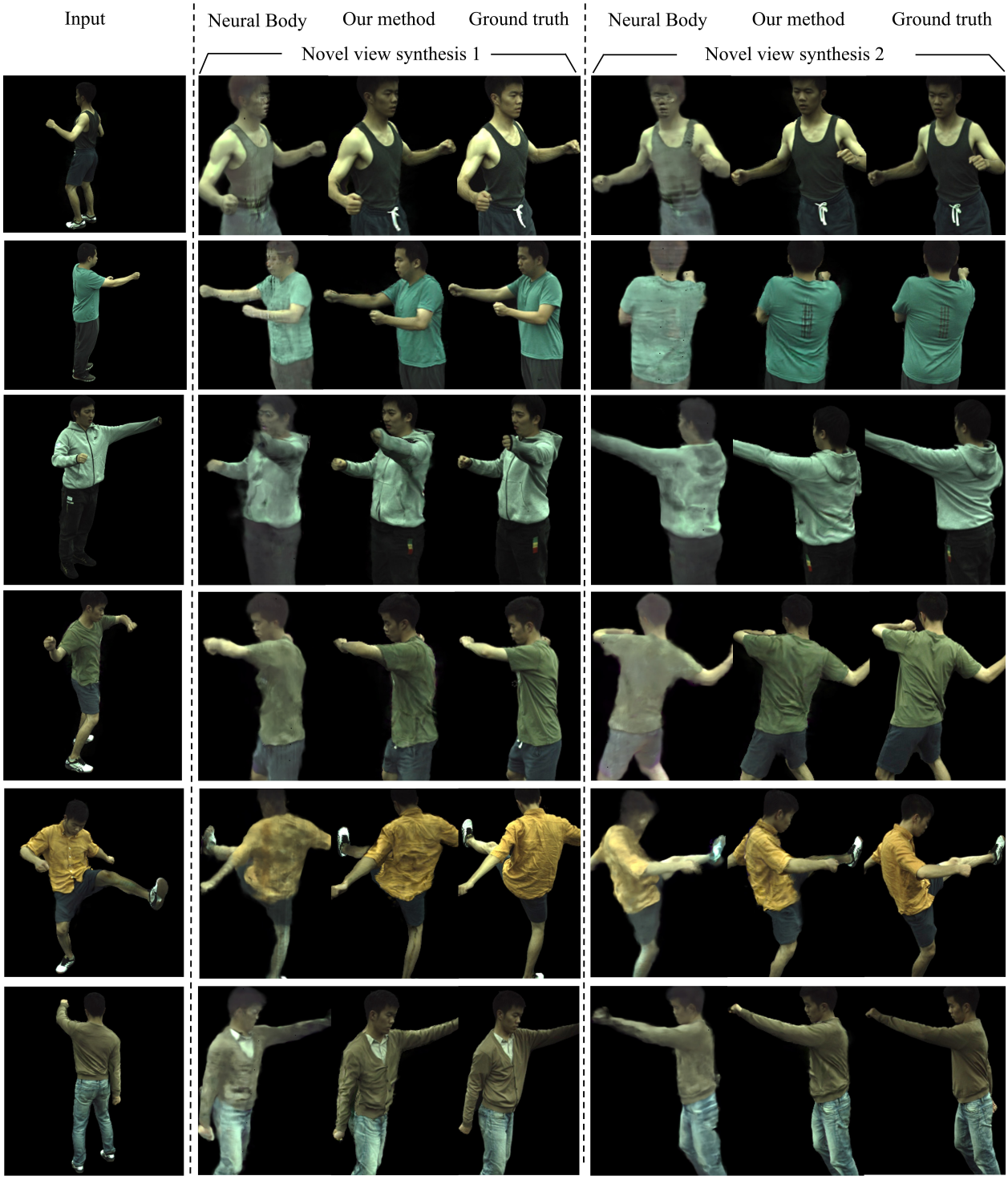


Figure 4.4: Qualitative comparison on ZJU-MoCap dataset.

Figure 4.5: Qualitative comparison for self-captured videos (first two rows) and YouTube videos (bottom three).

4.5.5 Ablation studies

Non-rigid motion & pose correction Fig. 4.6 shows visually, for in-the-wild data, the importance of including non-rigid motion and, additionally, pose correction for an unseen view.

(a) input (b) w/o pose correction w/o non-rigid motion (c) w/o non-rigid motion (d) full model

Figure 4.6: Pose correction and non-rigid motion improve novel view synthesis. Pose correction straightens the right arm and adds details (red arrows in (b) vs (c)) and non-rigid deformation improves clothing alignment and shape (green arrows in (c) vs. (d)).

	PSNR ↑	SSIM ↑	LPIPS* ↓
Neural Body [126]	29.08	0.9616	52.27
Ours (w/o non-rigid)	29.81	0.9657	34.17
Ours (full model)	30.24	0.9679	31.73

Table 4.3: Ablation study on ZJU-MoCap. We compute averages over 6 sequences. We color cells with best best and second best metric values. LPIPS* = LPIPS $\times 10^3$.

Table 4.3 illustrates that skeletal deformation alone is enough for significant improvement over

Neural Body for the ZJU-MoCap data. Adding non-rigid deformation provides further gains. (Accurate poses were provided for this dataset, thus we did not perform an ablation for the pose optimizer here.)

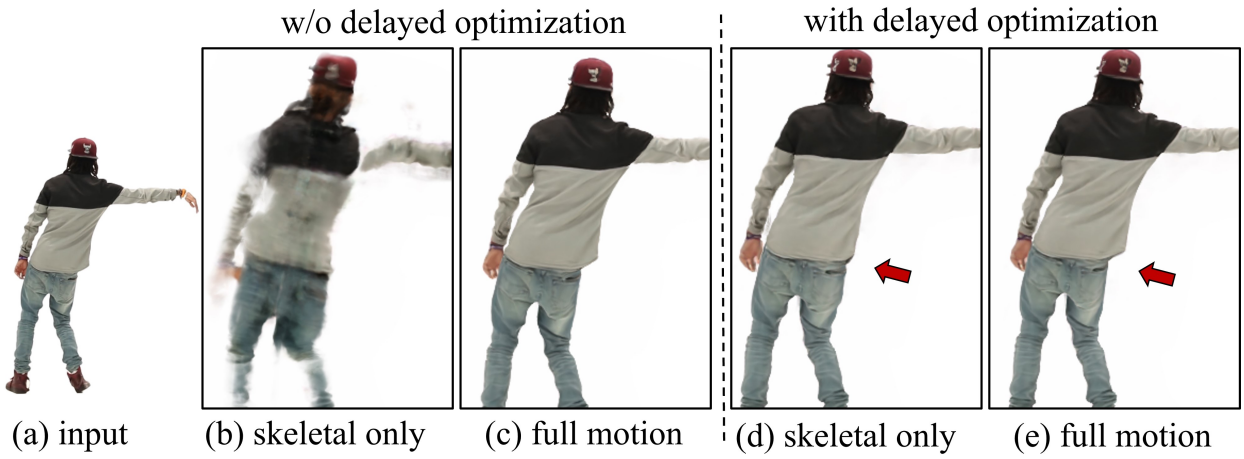


Figure 4.7: Delayed optimization (d, e) leads to better motion decoupling than the result without it (b, c). The skeletal-only deformation result without delayed optimization is poor, which can be "corrected" by the non-rigid deformation, but leads to poor view generalization (see Fig. 4.8).

Delayed optimization Fig. 4.7 shows the importance of delayed optimization for decoupling skeletal deformation and non-rigid deformation. When not decoupled well, generalization to new views is much poorer, as shown in Fig. 4.8.

(a) input (b) w/o delayed optimization (c) with delayed optimization

Figure 4.8: Without delayed optimization and strong decoupling of skeletal and non-rigid deformations, generalization to unseen views is poor (b). With delayed optimization, the decoupling leads to good generalization (c).

Sequence Length To understand how our method performs on different sequence lengths, we evaluate it on the sequences that vary in the number of frames but are sampled from the same video.

	PSNR ↑	SSIM ↑	LPIPS* ↓
556 frames	31.04	0.9705	32.12
228 frames	30.84	0.9701	31.78
112 frames	31.01	0.9703	32.75
56 frames	30.90	0.9693	35.45
19 frames	30.51	0.9655	45.17

Table 4.4: Ablation study on sequence length. We color cells with best metric values. LPIPS* = LPIPS $\times 10^3$.

Specifically, we take subject 392 from ZJU-MoCap dataset and use images captured from "camera 1" temporally sub-sampled at rates of 1, 2, 5, 10, and 30, yielding five training sequences containing 556, 228, 112, 56, and 19 frames respectively. For evaluation, we use the same motion sequence temporally sub-sampled by 30 but captured from the other 22 cameras not seen in the training. We use the same hyperparameters and training iterations throughout. The results are shown in Table 4.4.

As expected, using more frames leads to better quality; however the improvement is not obvious when the frame number is over a threshold (in this case, 112 frames). We speculate that diversity of body poses is a more significant factor in reconstruction quality than number of frames.

4.5.6 Optimized Canonical Appearance

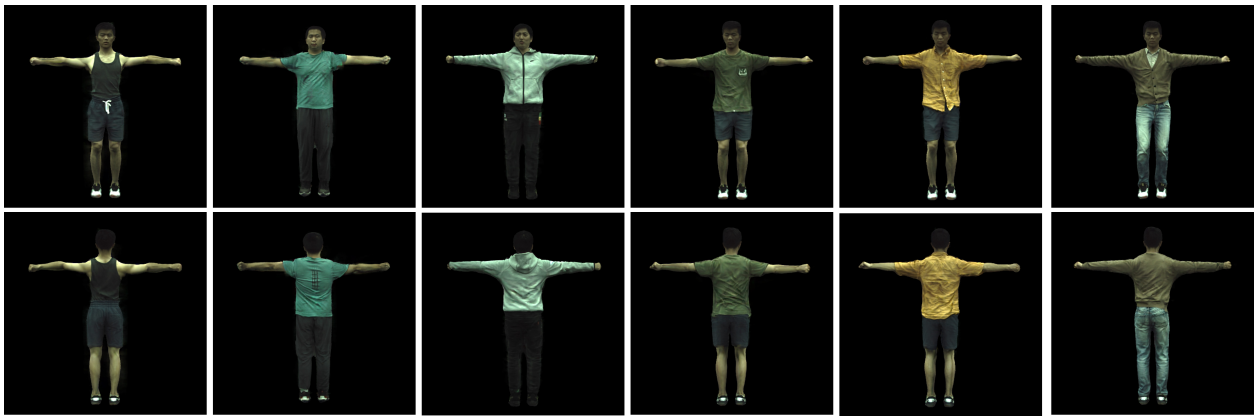


Figure 4.9: Optimized canonical appearance on ZJU-MoCap dataset.

Fig. 4.9 shows the recovered appearance for the pre-defined T-pose on the ZJU-MoCap [126] dataset; the results for self-captured and YouTube videos are shown in Fig. 4.10.

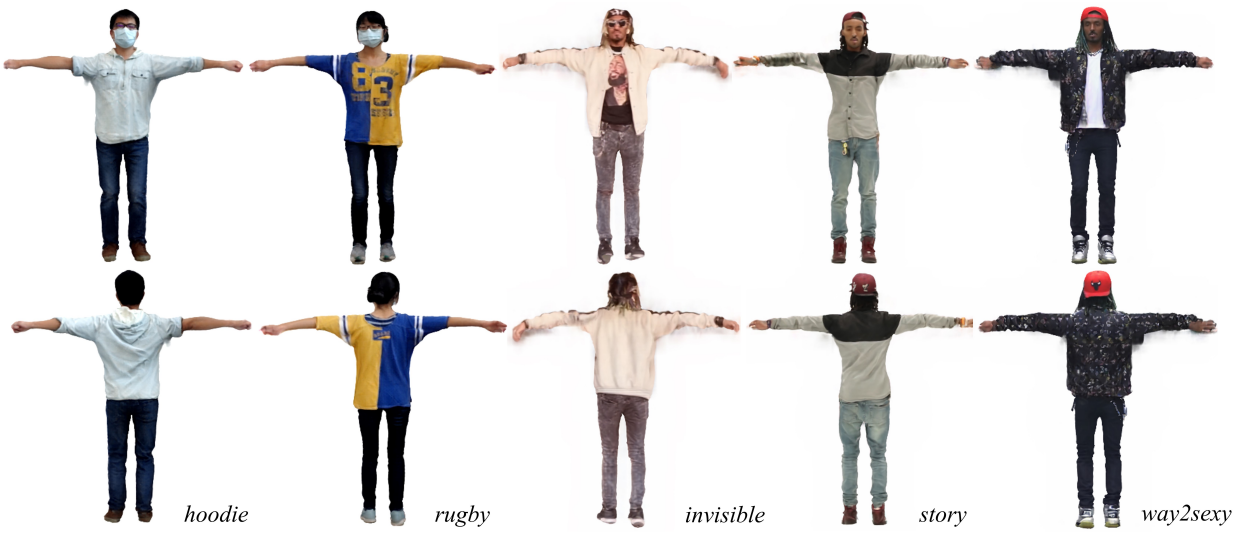


Figure 4.10: Optimized canonical appearance for self-captured videos (first two columns) and YouTube videos (right three).

4.6 Discussion

Limitations Our method has artifacts when part of the body is not shown in the video. Pose correction improves image alignment but may fail if the initial pose estimate is poor or if the image contains strong artifacts such as motion blur. In addition, we observed the frame-by-frame body poses are still not temporally smooth even after pose correction. We assume non-rigid motion is pose-dependent, but this is not always true (e.g., clothes shifting due to wind or due to follow-through after dynamic subject motion). We also assume fairly diffuse lighting, so that appearance does not change dramatically as the points on the subject rotate around. Finally, for in-the-wild videos, we rely on manual intervention to correct segmentation errors. These limitations point to a range of interesting avenues for future work.

We provide two visual examples of our method's limitations in Fig. 4.11. Pose correction may fail if the video frame contains artifacts, e.g., strong motion blur, as shown in (a) and (b). Non-rigid motion was not fully recovered in (c) and (d), as the movement of the jacket depended on the temporal dynamics of subject motion.

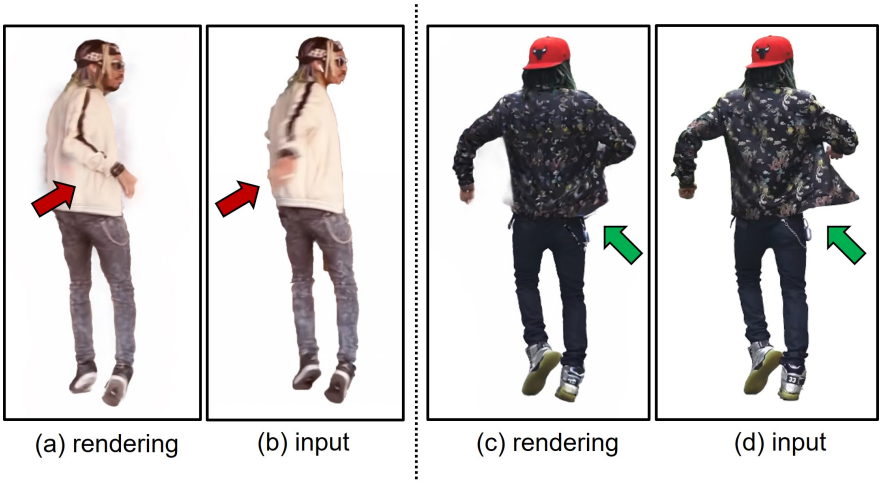


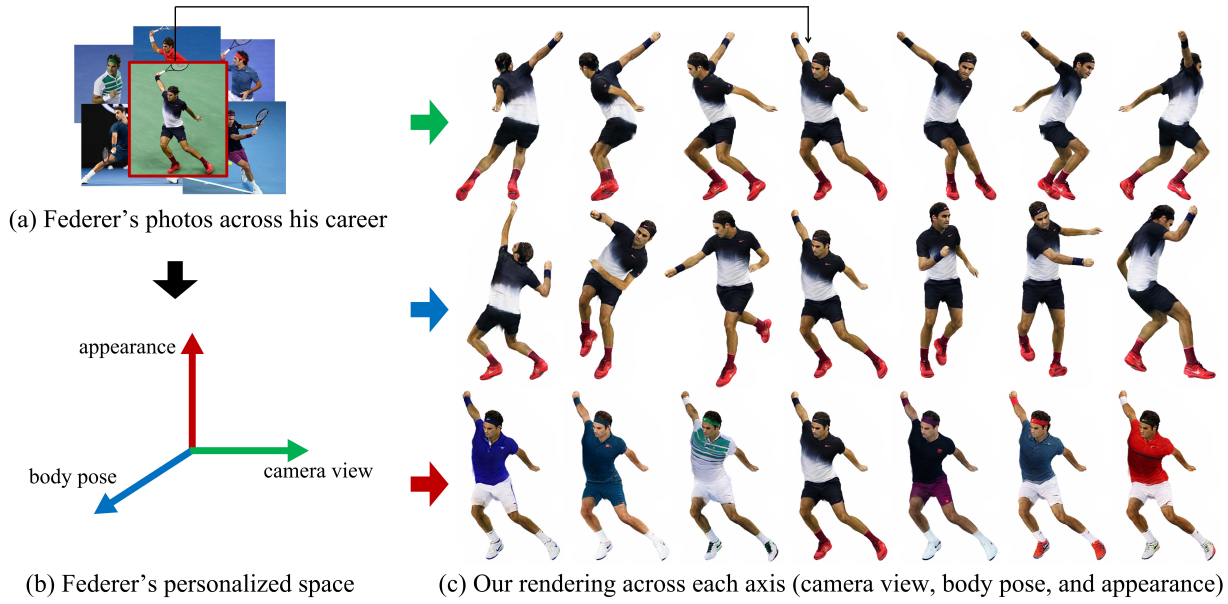
Figure 4.11: Visual examples of limitations. Pose correction may fail (a and b) and non-rigid clothes motion was not able to be fully recovered (c and d).

Societal Impact In this work we aim to faithfully reproduce motion sequences performed by a person with the capability of rendering unseen views. Therefore applying the technology to create false depictions, e.g., re-animating the subject in novel poses, was not considered as a potential application. Nevertheless, the public deployment of the technology should still be done with care, e.g., by reminding audiences that imagery is computer-generated when adjusting the viewpoint. In addition, the high computation requirement of the algorithm may lead to increased carbon emissions. We hope the methods that accelerate training of neural graphics primitives (e.g., [111]) will help reduce computation and thus the environmental impact. Finally, our method will be made available to the public for counter-measure analysis and computation reduction.

Conclusion We have presented HumanNeRF, producing state-of-the-art results for free-viewpoint renderings of moving people from monocular video. We demonstrate high fidelity results for this challenging scenario by carefully modeling body pose and motion as well as regularizing the optimization process. We hope the result points in a promising direction toward modeling humans in motion and, eventually, achieving fully photorealistic, free-viewpoint rendering of people from casual captures.

Chapter 5

PERSONNERF: PERSONALIZED RECONSTRUCTION FROM PHOTO COLLECTIONS



(a) Federer's photos across his career

(b) Federer's personalized space

(c) Our rendering across each axis (camera view, body pose, and appearance)

Figure 5.1: Given a photo collection of a subject (e.g., Roger Federer) (a), our method PersonNeRF builds a space of the subject spanned by camera view, body pose, and appearance (b). PersonNeRF enables traversing this space and exploring unobserved combinations of these attributes (c). Here we render novel views (top row), various body poses (middle row), and different appearances (bottom row) by traversing the corresponding axes. Among all of the renderings shown here, only the central images of rows correspond to a photo actually observed in the training data (corresponding input photo marked with a red square). *Photo credits to Getty Images.*

In this chapter, I present a collaborative research project involving Pratul P. Srinivasan, Brian Curless, and Ira Kemelmacher-Shlizerman. The work will be published in CVPR 2023 [179].

5.1 Introduction

We present a method for transforming an unstructured personal photo collection, containing images spanning multiple years with different outfits, appearances, and body poses, into a 3D representation of the subject. Our system, which we call PersonNeRF, enables us to render the subject under novel unobserved combinations of camera viewpoint, body pose, and appearance.

Free-viewpoint rendering from unstructured photos is a particularly challenging task because a photo collection can contain images at different times where the subject has different clothing and appearance. Furthermore, we only have access to a handful of images for each appearance, so it is unlikely that all regions of the body would be well-observed for any given appearance. In addition, any given body pose is likely observed from just a single or very few camera viewpoints.

We address this challenging scenario of sparse viewpoint and pose observations with changing appearance by modeling a single canonical-pose neural volumetric representation that uses a shared motion weight field to describe how the canonical volume deforms with changes in body pose, all conditioned on appearance-dependent latent vectors. Our key insight is that although the observed body poses have different appearances across the photo collection, they should all be explained by a common motion model since they all come from the same person. Furthermore, although the appearances of a subject can vary across the photo collection, they all share common properties such as symmetry so embedding appearance in a shared latent space can help the model learn useful priors.

To this end, we build our work on top of HumanNeRF [178], which is a state-of-the-art free-viewpoint human rendering approach that requires hundreds of images of a subject without clothing or appearance changes. Along with regularization, we extend HumanNeRF to account for sparse observations as well as enable modeling diverse appearances. Finally, we build an entire personalized space spanned by camera view, body pose, and appearance that allows intuitive exploration of arbitrary novel combinations of these attributes (as shown in Fig. 5.1). **Please refer to the project**

page[1] for video demonstrations and interactive demos.

5.2 Related Work

3D reconstruction from unstructured photos Reconstructing static scenes from unstructured photo collections is a longstanding research problem in the fields of computer vision and graphics. The seminal Photo Tourism system [151] applies large-scale structure-from-motion [144] to tourist photos of famous sites, enabling interactive navigation of the 3D scene. Subsequent works leveraged multi-view stereo [147, 42] to increase the 3D reconstruction quality [148, 1]. Recently, this problem has been revisited with neural rendering [162, 163, 108, 89, 156]. In particular, Neural Radiance Fields (NeRFs) [110] have enabled photorealistic view synthesis results of challenging scenes, including tourist sites [103] and even city-scale scenes [160]. In addition to static scenes, unstructured photo collections have been also used to model human faces [73, 90] or even visualize scene changes through time [100, 101, 106].

Our method builds on top of NeRF's neural volumetric representation of static scenes, and extends it to model dynamic human bodies from unstructured photo collections.

3D reconstruction of humans Many early works in image-based rendering [158] have addressed the task of rendering novel views of human bodies. These techniques are largely based on view-dependent texture mapping [32], which reprojects observed images into each novel viewpoint using a proxy geometry. The image-based rendering community has explored many geometry proxies for rendering humans, including depth maps [198, 69], visual hulls [105], and parametric human models [18]. An alternative technique for 3D reconstruction and rendering of humans is to use 3D scanning techniques to recover a signed distance field representation [29, 36], and then extract and texture a polygon mesh [48, 26, 102]. Recently, neural field representations [184], have become popular for modeling humans since they are suited for representing surfaces with arbitrary topology. Methods have reconstructed neural field representations of humans from a variety of different inputs, including 3D scans[165, 23, 99, 109, 138], multi-view RGB observations[126, 85, 92],

[1] https://grail.cs.washington.edu/projects/personnerf/

RGB-D sequences[35], or monocular videos [178, 67]. Our work is most closely related to HumanNeRF [178], which reconstructs a volumetric neural field from a monocular video of a moving human. We build upon this representation and extend it to enable reconstructing a neural volumetric model from unstructured photo collections with diverse poses and appearances.

5.3 Method

In this section, we first review HumanNeRF [178] (Sec. 5.3.1), explain how we regularize it to improve reconstruction from sparse inputs (Sec. 5.3.2), and then describe how we model diverse appearances (Sec. 5.3.3 and 5.3.4). Finally, we describe how we build a personalized space to support intuitive exploration (Sec. 5.3.5).

5.3.1 Background

HumanNeRF The recently-introduced HumanNeRF method represents a moving person as a canonical volume F_c warped to a body pose $\mathbf{p}$ to produce a volume F_o in observed space:

$$F_o(\mathbf{x}, \mathbf{p}) = F_c(T(\mathbf{x}, \mathbf{p})), \tag{5.1}$$

where $T : (\mathbf{x}_o, \mathbf{p}) \rightarrow \mathbf{x}_c$ defines a motion field mapping points from observed space back to canonical space, and $F_c : \mathbf{x} \rightarrow (\mathbf{c}, \sigma)$ maps position $\mathbf{x}$ to color $\mathbf{c}$ and density σ, represented by $\mathrm{MLP}_{\theta_c}(\gamma(\mathbf{x}))$ taking $\gamma(\mathbf{x})$, a sinusoidal positional encoding of $\mathbf{x}$, as input, with parameters θ_c.

The motion field T is further decomposed into skeletal motion T_{skel} and non-rigid motion T_{NR}:

$$T(\mathbf{x}, \mathbf{p}) = T_{\text{skel}}(\mathbf{x}, P_{\text{pose}}(\mathbf{p})) + T_{\text{NR}}(\mathbf{x}_{\text{skel}}, \mathbf{p}), \tag{5.2}$$

where $\mathbf{x}_{\text{skel}} = T_{\text{skel}}(\mathbf{x}, P_{\text{pose}}(\mathbf{p}))$, T_{NR} represented by $\mathrm{MLP}_{\theta_{\text{NR}}}$ predicts a non-rigid offset $\Delta\mathbf{x}$, and $P_{\text{pose}}(\mathbf{p})$ corrects the body pose $\mathbf{p} = (J, \Omega)$ with the residual of joint angles Δ_Ω predicted by $\mathrm{MLP}_{\theta_{\text{pose}}}(\Omega)$ taking joint angles Ω as input.

The skeletal motion T_{skel} maps an observed position to the canonical space, computed as a weighted sum of K motion bases $(R_i, \mathbf{t}_i)$:

$$T_{\text{skel}}(\mathbf{x}, \mathbf{p}) = \sum_{i=1}^{K} w_o^i(\mathbf{x})(R_i\mathbf{x} + \mathbf{t}_i), \tag{5.3}$$

where $(R_i, \mathbf{t}_i)$, explicitly computed from $\mathbf{p}$, indicates the rotation and translation that maps i-th bone from observation to canonical space and w_o^i is the corresponding weight in observed space.

Each w_o^i is approximated using weights w_c^i defined in canonical space:

$$w_o^i(\mathbf{x}) = \frac{w_c^i(R_i\mathbf{x} + \mathbf{t}_i)}{\sum_{k=1}^{K} w_c^k(R_k\mathbf{x} + \mathbf{t}_k)}. \tag{5.4}$$

HumanNeRF stores the set of $\{w_c^i(\mathbf{x})\}$ and a background class into a single volume grid $W_c(\mathbf{x})$ with $K + 1$ channels, generated by a convolution network $\mathrm{CNN}_{\theta_{\text{skel}}}$ that takes as input a random (constant) latent code $\mathbf{z}$.

Volume Rendering The observed volume F_o that produces color $\mathbf{c}$ and density σ is rendered using the volume rendering equation [110]. The expected color $\mathbf{C}(\mathbf{r})$ of a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ with G samples is computed as:

$$\begin{aligned} \mathbf{C}(\mathbf{r}) &= \sum_{i=1}^{G}(\prod_{j=1}^{i-1}(1 - \alpha_j))\alpha_i\mathbf{c}(\mathbf{x}_i), \\ \alpha_i &= f(\mathbf{x}_i)(1 - \exp(-\sigma(\mathbf{x}_i)\Delta t_i)), \end{aligned} \tag{5.5}$$

where $\Delta t_i = t_{i+1} - t_i$ is sample interval, and $f(\mathbf{x}) = \sum_{k=1}^{K} w_c^k(R_k\mathbf{x} + \mathbf{t}_k)$ is foreground likelihood. Finally, HumanNeRF optimizes for network parameters $\Theta = \{\theta_c, \theta_{\text{skel}}, \theta_{\text{NR}}, \theta_{\text{pose}}\}$ through MSE loss, $\mathcal{L}_{\text{MSE}}$, and LPIPS [193] loss, $\mathcal{L}_{\text{LPIPS}}$, by comparing renderings with inputs.

Our method Built upon HumanNeRF, our method addresses the difficulties of insufficient observations and a variety of subject appearances. In the following sections, we detail our approaches to overcoming these challenges. The overview of our method is shown in Fig. 5.2.

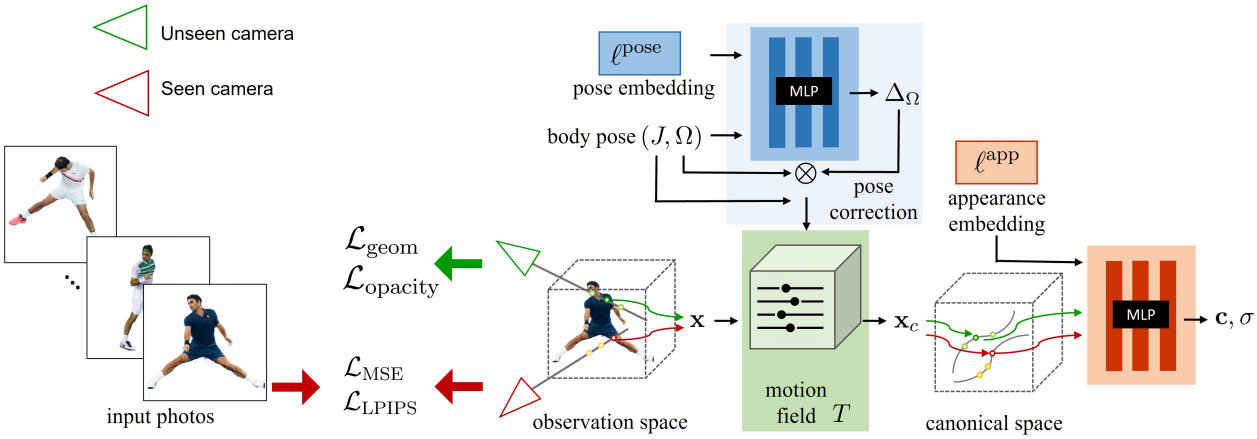


Figure 5.2: Given an input personal photo collection, our method optimizes for a canonical volume that can render diverse appearances. We represent the canonical volume with an MLP conditioned on an appearance embedding, and use a shared pose-dependent motion field that maps from observation to canonical space. Additionally, we use a pose correction MLP that takes the estimated body pose and a pose embedding and outputs appearance-dependent pose residuals. Finally, to improve rendering quality from sparse observations, we regularize the volumetric representation to have smooth and opaque geometry with $\mathcal{L}_{\text{geom}}$ and $\mathcal{L}_{\text{opacity}}$, which we apply to renderings from uniformly-sampled unobserved camera viewpoints. *Photo credits to Getty Images.*

5.3.2 Unseen view regularization

Although HumanNeRF [178] works well given monocular videos, we observe it produces poor results on unstructured photo collections due to insufficient observations: we usually only have a handful of photos of a subject's outfit ($<$ 25 images in our case) while HumanNeRF relies on videos with a large number of video frames ($>$ 300 frames).

We find HumanNeRF's struggles in our setting for two reasons: (1) its non-rigid motion does not generalize well to novel viewpoints since there are too few pose observations to sufficiently constrain this pose-dependent effect; (2) the reconstructed canonical-pose human body geometry is incorrect due to insufficient viewpoint observations, resulting in inconsistent appearance in rendered novel viewpoints.

We address the first limitation by simply removing the non-rigid component and only use skeletal motion:

$$T(\mathbf{x}, \mathbf{p}) = T_{\text{skel}}(\mathbf{x}, P_{\text{pose}}(\mathbf{p})) \tag{5.6}$$

We address the second limitation by regularizing the body geometry as rendered in novel views. Specifically, inspired by RegNeRF [114], we encourage the geometry to be smooth by enforcing a depth smoothness loss on rendered depth maps. We generate novel camera poses by first sampling an angle ϕ from a uniform distribution, $\phi \sim U(0, 2\pi)$, and rotate the input camera with ϕ around the up vector with respect to the body center.

We render a pixel's depth value by calculating the expected ray termination position, using the same volume rendering weights used to compute the pixel's color (Eq. 5.5):

$$D(\mathbf{r}) = \sum_{i=1}^{G} (\prod_{j=1}^{i-1} (1 - \alpha_j)) \alpha_i t_i. \tag{5.7}$$

Likewise, we compute a pixel's alpha value as:

$$A(\mathbf{r}) = \sum_{i=1}^{G} (\prod_{j=1}^{i-1} (1 - \alpha_j)) \alpha_i. \tag{5.8}$$

Our proposed depth smoothness loss is formulated as:

$$\begin{aligned} \mathcal{L}_{\text{geom}} = \sum_{i,j=1}^{H-1} & \left(A(\mathbf{r}_{i,j}) A(\mathbf{r}_{i,j+1}) (D(\mathbf{r}_{i,j}) - D(\mathbf{r}_{i,j+1}))\right)^2 \\ & + \left(A(\mathbf{r}_{i,j}) A(\mathbf{r}_{i+1,j}) (D(\mathbf{r}_{i,j}) - D(\mathbf{r}_{i+1,j}))\right)^2 . \end{aligned} \tag{5.9}$$

where the loss is evaluated over patches of size H, as we use patch-based ray sampling similar to HumanNeRF. Note that this loss only penalizes depth discontinuities when the alphas of neighboring points are high, which effectively constrains the loss to points on the surface.

In practice, we find the depth smoothness term improves geometry and rendering but introduces "haze" artifacts around the subject. This problem arises because the loss encourages small alphas – all zero alpha would in fact minimize this term – biasing toward transparent geometry.

To address this problem, we use an opacity loss inspired by Neural Volumes [94] that encourages binary alphas:

$$\mathcal{L}_{\text{opacity}} = \sum_{i,j} \log(A(\mathbf{r}_{i,j}) + \epsilon) + \log(1 - A(\mathbf{r}_{i,j}) + \epsilon) - C, \tag{5.10}$$

where $C = \log(\epsilon) + \log(1 + \epsilon)$ to ensure non-negativity.

5.3.3 Appearance modeling

We take as input photos of a subject taken at different times; these photos are subdivided into *appearance sets* corresponding to photos taken around the same time, i.e., with the same clothing, etc.

When modeling diverse appearances of a subject, we want to achieve two goals: (1) **appearance consistency**: synthesizing consistent texture in unobserved regions in one appearance set with the help of the others; (2) **pose consistency**: a motion model that keeps the rendered pose consistent when switching the subject's appearance.

A naive approach is to train a separate network on each appearance set. This approach does not perform well: (1) the canonical MLP sees very few images in the training, resulting in artifacts in unobserved regions, thus degrading appearance consistency (Fig. 5.5-(a)); (2) the learned motion weight volume overfits body poses in each (small) appearance set and does not generalize well to the other sets, leading to poor pose consistency (Fig. 5.5-(b)).

Instead, we propose to train all photos with different appearances into a single network. Specifically, we enforce the shared canonical appearance MLP_{θ_c} to be appearance-dependent but optimize for a single, universal motion weight volume W_c across all images. The shared, appearance-conditioned canonical MLP synthesizes consistent textures by generalizing over the full set of images seen in training, while the universal motion weight volume significantly improves pose consistency, as it is trained on the full set of body poses.

To condition the canonical MLP, inspired by Martin-Brualla et al. [103], we adopt the approach of Generative Latent Optimization [15], where each appearance set (with index i) is bound to a single real-valued appearance embedding vector $\ell^{\text{app}}_{(i)}$. This vector is concatenated with $\gamma(\mathbf{x})$ as

input to the canonical MLP_{θ_c}. As a result, the canonical volume F_c is appearance-dependent:

$$F_c(\mathbf{x}, \ell_{(i)}^{\text{app}}) = \mathrm{MLP}_{\theta_c}(\gamma(\mathbf{x}), \ell_{(i)}^{\text{app}}). \tag{5.11}$$

Similarly, we introduce pose embedding vector $\ell_{(i)}^{\text{pose}}$ to condition the pose correction module on each appearance set and concatenate this vector with Ω as input to $\mathrm{MLP}_{\theta_{\text{pose}}}$. By doing so, we can differentiate between instances where two bodies have an identical pose but different clothing.

The appearance embeddings $L^{\text{app}} = \{\ell_{(i)}^{\text{app}}\}_{i=1}^{S}$ as well as pose embeddings $L^{\text{pose}} = \{\ell_{(i)}^{\text{pose}}\}_{i=1}^{S}$ are optimized alongside other network parameters, where S is the number of appearance sets.

5.3.4 Optimization

Loss function Our total loss is a combination of the previously-discussed losses:

$$\mathcal{L} = \mathcal{L}_{\text{LPIPS}} + \lambda_1 \mathcal{L}_{\text{MSE}} + \lambda_2 \mathcal{L}_{\text{geom}} + \lambda_3 \mathcal{L}_{\text{opacity}}. \tag{5.12}$$

Objective Given input images $\{I_1, I_2, ..., I_N\}$, appearance set indices $\{s_1, s_2, ..., s_N\}$, body poses $\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\}$, and cameras $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N\}$, we optimize the objective:

$$\underset{\Theta}{\text{minimize}} \quad \sum_{i=1}^{N} \mathcal{L}(\Gamma[F_c(T(\mathbf{x}, \mathbf{p}_i, \ell_{(s_i)}^{\text{pose}}), \ell_{(s_i)}^{\text{app}}), \mathbf{e}_i], I_i), \tag{5.13}$$

where $\mathcal{L}(\cdot)$ is the loss function and $\Gamma[\cdot]$ is a volume renderer, and we minimize the loss with respect to all network parameters and embedding vectors $\Theta = \{\theta_c, \theta_{\text{skel}}, \theta_{\text{pose}}, L^{\text{app}}, L^{\text{pose}}\}$.

We shoot rays toward both seen and unseen cameras. $\mathcal{L}_{\text{LPIPS}}$ and $\mathcal{L}_{\text{MSE}}$ are computed from the output of seen cameras, while $\mathcal{L}_{\text{geom}}$ and $\mathcal{L}_{\text{opacity}}$ are applied to renderings of unseen ones. We use $\lambda_1 = 0.2$, $\lambda_2 = 1.0$, and $\lambda_3 = 10.0$. Additionally, we stop the gradient flow through the pose MLP when backpropagating $\mathcal{L}_{\text{geom}}$, as we found it can lead to degenerate pose correction.

5.3.5 Building a personalized space

Once the optimization converges, we use its result to build a personalized space of the subject spanned by camera view, body pose, and appearance. We allow continuous variation in viewpoint,

but restrict body pose and appearance to those that were observed in the set. Every point in the space has a corresponding rendering.

In practice, the space is defined as a cube with size 1 where the coordinate value ranges from 0 to 1. Our goal is to map a point in that cube to the inputs of the network from which we render the subject.

Specifically, assuming the subject has N body poses and S appearances, we need to perform mapping on coordinates (a, b, c) corresponding to position along the axes of appearance, body pose, and camera view, respectively:

(1) **Appearances**: we map the value a to the index of S appearances: $ia = \lfloor aS \rfloor$, which was used to retrieve the appearance embedding $\ell^{\text{app}}_{(ia)}$ for canonical MLP_{θ_c}.

(2) **Body pose**: we map the value b to the index of N body poses: $ib = \lfloor bN \rfloor$. We get the ib -th body pose $\mathbf{p}$, corresponding to appearance index s_{ib}. We then take pose embedding $\ell^{\text{pose}}_{(s_{ib})}$ as input for pose $\text{MLP}_{\theta_{\text{pose}}}$.

(3) **Camera view**: we rotate the camera $\mathbf{e}_{ib}$ by $\phi = 2\pi c$ around up vector with respect to the body center to get a viewing camera $\mathbf{e}_{\text{v}}$.

Finally, we generate a subject rendering corresponding to the position (a, b, c) by feeding the appearance embedding $\ell^{\text{app}}_{(ia)}$, pose embedding $\ell^{\text{pose}}_{(s_{ib})}$, and body pose $\mathbf{p}$ to the network and producing a volume in observation space rendered by the viewing camera $\mathbf{e}_{\text{v}}$.

5.4 Results

5.4.1 Dataset

In this section, we include the results of experiments using a photo collection of Roger Federer. The results on more subjects, including Rafael Nadal, Noval Djokovic, and Serena Willaims, are shown in Sec. 5.6.

The Roger Federer dataset contains 10 appearance sets spanning 12 years. We collect photos by searching for a specific game in a particular year (e.g., "2019 Australian Open Final"). We collected 19 to 24 photos for each game, one per year, and label each set according to the year

(*2009*, *2012*, ..., *2020*).

Following HumanNeRF [178], we run SPIN [80] to estimate body pose and camera pose, automatically segment the subject, and manually correct segmentation errors and 3D body poses with obvious errors. Additionally, for images where the subject is occluded by balls or rackets, we label the regions of occluded objects and omit them during optimization.

5.4.2 Implementation details

We optimize Eq. 4.11 using the Adam optimizer [78] with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We set the learning rate to 5×10^{-4} for θ_c (the canonical MLP), L^{app}, and L^{pose} (embedding vectors), and 5×10^{-5} for all the others. We sample 128 points along each ray for rendering. The size of embedding vectors of ℓ^{app} and ℓ^{pose} are 256 and 16. We use patch-based ray sampling with 6 patches with size 32x32 for seen cameras and 16 patches with size 8x8 for unseen ones. The optimization takes 200K iterations to converge when training each game with individual networks and takes 600K iterations for all games into a single network. Additionally, we delay pose refinement, geometry regularization, and opacity constraint until after 1K, 1K, and 50K iterations for separate-networks training, and 1K, 10K, and 200K iterations for single-network optimization.

5.4.3 Comparison

We compare our method with HumanNeRF [178], the state-of-the-art free-viewpoint method on monocular videos. We run experiments on individual datasets (*2009*, *2012*, ..., *2020*). We use the official HumanNeRF implementation with hyperparameters $T_s = 2.5K$ and $T_e = 5K$ to accommodate the much smaller input dataset size. Because HumanNeRF only can optimize for a single appearance, we do the same in our method. Finally, we train HumanNeRF with 200K iterations, the same number used in our method.

Evaluation protocol As we lack ground truth when evaluating results rendered from unseen views, we adopt Frechet inception distance (FID) [58] for quantitative comparison. For each input image, we rotate the camera in 10-degree increments around the "up" vector w.r.t the body center and use these renderings for evaluation.

	2009	2012	2013	2014	2015	2016	2017	2018	2019	2020
HumanNeRF	70.64	80.62	75.09	73.00	93.89	83.35	82.19	69.40	67.47	73.01
Our method	59.28	63.92	68.92	63.39	77.36	71.99	71.98	58.38	58.21	61.77

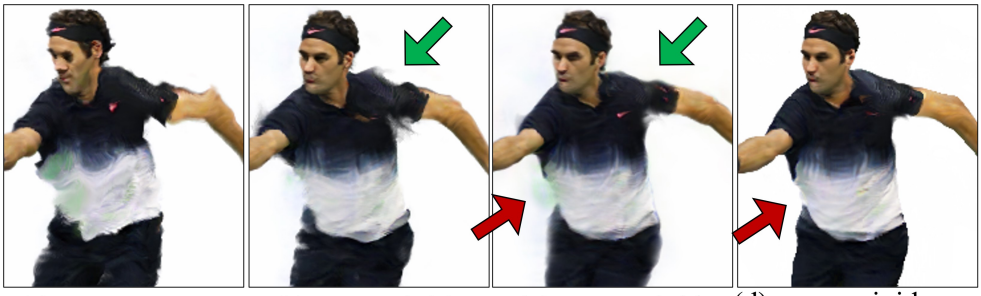
Table 5.1: Comparison to HumanNeRF [178]: FID is computed per dataset (per year). Lower FID score is better. We color cells that have the best metric value. Our method outperforms HumanNeRF with significant margins.

Results Quantitatively, as shown in Table 5.1, our method outperforms HumanNeRF on all datasets by comfortable margins. The performance gain is particularly significant when visualizing the results, as shown in Fig. 5.3. Our method is able to create consistent geometry, sharp details, and nice renderings, while HumanNeRF tends to produce irregular shapes, distorted textures, and noisy images, due to insufficient inputs.

Visualization of Roger Federer's space In Fig. 5.6, we visualize the rebuilt Federer space by keeping the body pose fixed and rendering dense samples in the camera-appearance plane starting from one photo. In this case, only a single image (the one with a red square) is directly observed, showing how sparse observations we have to rebuild the space. The renderings are sharp and with few artifacts, and the appearance and pose consistency are well-maintained. More visualizations of personalized space on Roger Federer and other subjects are shown in Sec. 5.6.

Ablation studies Fig. 5.4 shows visually how we outperform HumanNeRF by modifying the model and introducing new losses. By removing non-rigid motion, we get a significant quality boost. We further enhance the shape and texture reconstruction with the geometry and opacity losses. Table 5.2 quantifies the importance of each element. We get the best performance when including all the refinements.

Figure 5.3: Our method produces more convincing renderings with fewer artifacts than those from HumanNeRF [178]. Note how HumanNeRF produces errors in regions occluded from the input view, while our method produces plausible geometry. *Photo credits to Getty Images.*



(a) HumanNeRF (b) – non-rigid (c) – non-rigid + geom. (d) – non-rigid + geom., opacity

Figure 5.4: Ablation study. Removing the non-rigid motion component from HumanNeRF significantly improves reconstruction quality. Adding our geometry loss further refines the shape (green arrow) but introduces "haze" artifacts (red arrow), which we address with the opacity loss.

	FID ↓
HumanNeRF [178]	76.87
Ours – non-rigid	71.75
Ours – non-rigid + geometry	76.84
Ours – non-rigid + opacity	67.01
Ours + geometry, opacity	65.91
Ours – non-rigid + geometry, opacity	**65.52**

Table 5.2: Ablation: average FID (lower is better) over 10 datasets. We color cells with best and second best metric values.

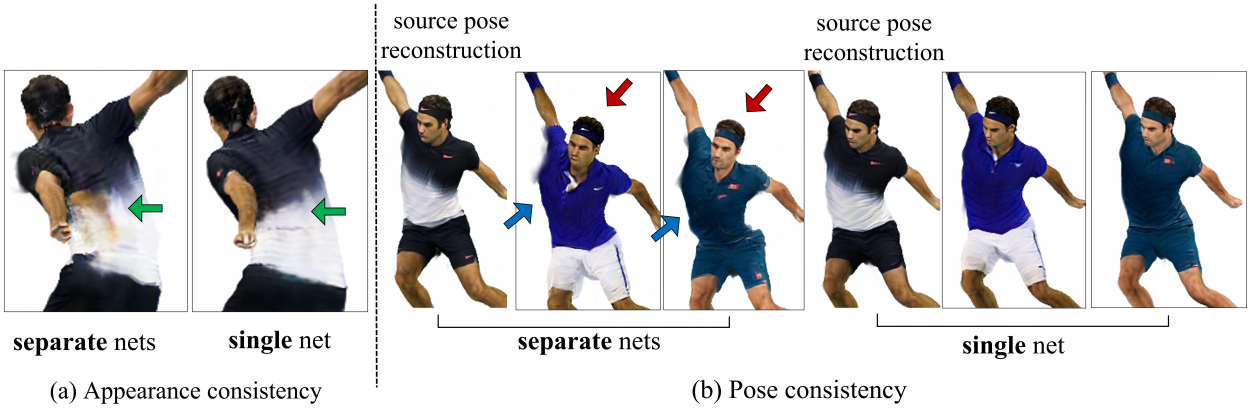


Figure 5.5: (a) **Appearance consistency**: training all appearance sets with a single network synthesizes higher quality texture for unobserved regions while training with separate networks produces incompatible colors (green arrow). (b) **Pose consistency**: In comparison to the source pose reconstruction (i.e., the combination of pose and appearance is observed in training), separate-networks training produces unsatisfied results when combining the pose with unseen appearances; the head orientations are different from the input (red arrow) and the bodies are unnaturally distorted (blue arrow). In contrast, single-network optimization enables consistent output.

Appearance and pose consistency Fig. 5.5 illustrates the benefit of training all images with a single network. In contrast to individually trained networks, Fig. 5.5-(a) illustrates it can synthesize compatible textures for unobserved regions as a result of better generalization, thus maintaining appearance consistency; Fig. 5.5-(b) demonstrates the unified network is able to keep the rendered body pose persistent across different appearances, thanks to the shared motion weight volume, hence guaranteeing pose consistency.

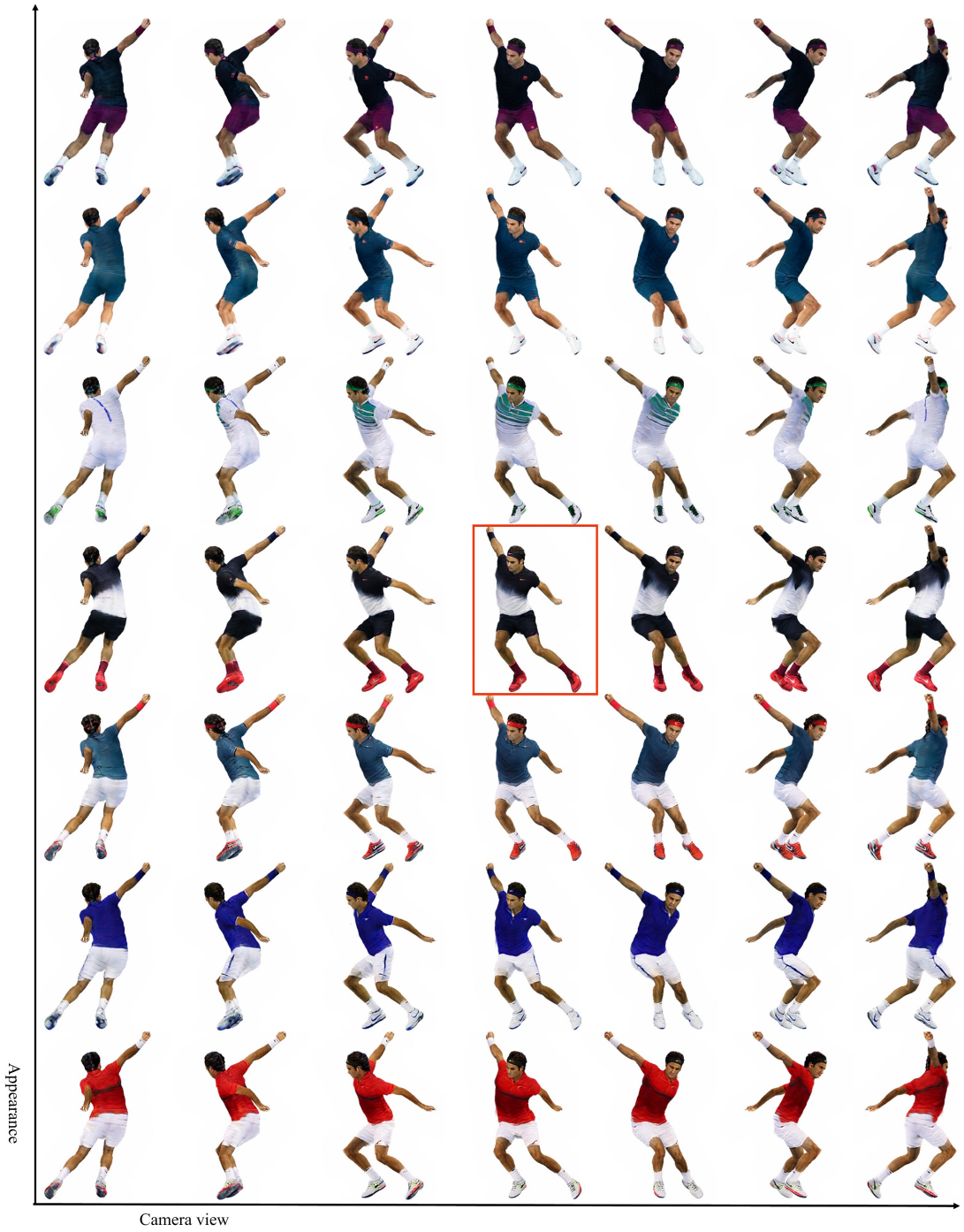


Figure 5.6: The visualization of the (appearance, camera view) plane of the reconstructed Federer space. Note that only the image in the red square was directly observed in the input data.

5.5 Discussion

Limitations Our work builds upon HumanNeRF to account for sparse inputs and diverse appearance. While it is effective in this challenging scenario, it inherits some of HumanNeRF's limitations such as its reliance on the initialized poses, its assumption of relatively diffuse lighting, and its requirement for manual human segmentation. Additionally, since human body pose estimators typically fail on images with heavily-occluded bodies, we can only use input photos that view the full body.

Societal impact In this work, we aim to faithfully produce images of a person with the capability of just rendering unseen views and switching appearance within their own set of appearances. The work does not intend to create motions and animations that didn't happen. While we show examples of our results on a variety of persons with different genders and skin tones, it is important to validate in future work that the method scales to a wide range of subjects.

Conclusion We have presented PersonNeRF, allowing rendering a human subject with arbitrary novel combinations of body pose, camera view, and appearance from an unstructured photo collection. Our method enables exploring these combinations by traversing a reconstructed space spanned by these attributes and demonstrates high-quality and consistent results across novel views and unobserved appearances.

5.6 More Results

In this section, other than Roger Federer, we demonstrate our method on a wide variety of subjects that cover different genders and skin tones. In particular, we show results on three tennis athletes, Novak Djokovic, Serena Williams, and Rafael Nadal where each has three appearance sets in the datasets we collected. We present quantitative results in FID in Table 5.3 and visually compare them with HumanNeRF [178] in Fig. 5.7. The quality improvement over the related work is similar to the case of Roger Federer.

	Novak Djokovic			**Serena Willams**			**Rafael Nadal**		
	2013	**2016**	**2019**	**2009**	**2010**	**2011**	**2014**	**2019**	**2022**
HumanNeRF	87.07	62.01	64.17	104.23	100.52	113.41	90.04	64.95	76.68
Our method	81.38	57.14	58.74	87.81	90.70	85.17	80.95	62.75	61.71

Table 5.3: Comparison to HumanNeRF [178]: FID is computed per subject per year. Lower FID score is better.

More visualizations of personalized space In Sec. 5.4, we show a visualization of (appearance, camera view) plane of the reconstructed space of Roger Federer. Here we show the other two planes, (appearance, body pose) plane in Fig. 5.8 and (body pose, camera view) plane in Fig. 5.9 where we keep the camera view and appearance fixed, respectively.

Additionally, we show visualizations of the rebuilt personalized spaces of the other 3 persons, Novak Djokovic in Fig. 5.10, 5.11 and 5.12, Serena Willams in Fig. 5.13, 5.14 and 5.15, and Rafael Nadal in Fig. 5.16, 5.17 and 5.18.

Figure 5.7: Visual comparisons to HumanNeRF [178] on the tennis athletes: Novak Djokovic, Serena Willams, and Rafael Nadal. *Photo credits to Getty Images.*

Figure 5.8: Visualization of the (appearance, body pose) plane of the reconstructed space of Roger Federer.

Figure 5.9: Visualization of the (body pose, camera view) plane of the reconstructed space of Roger Federer.

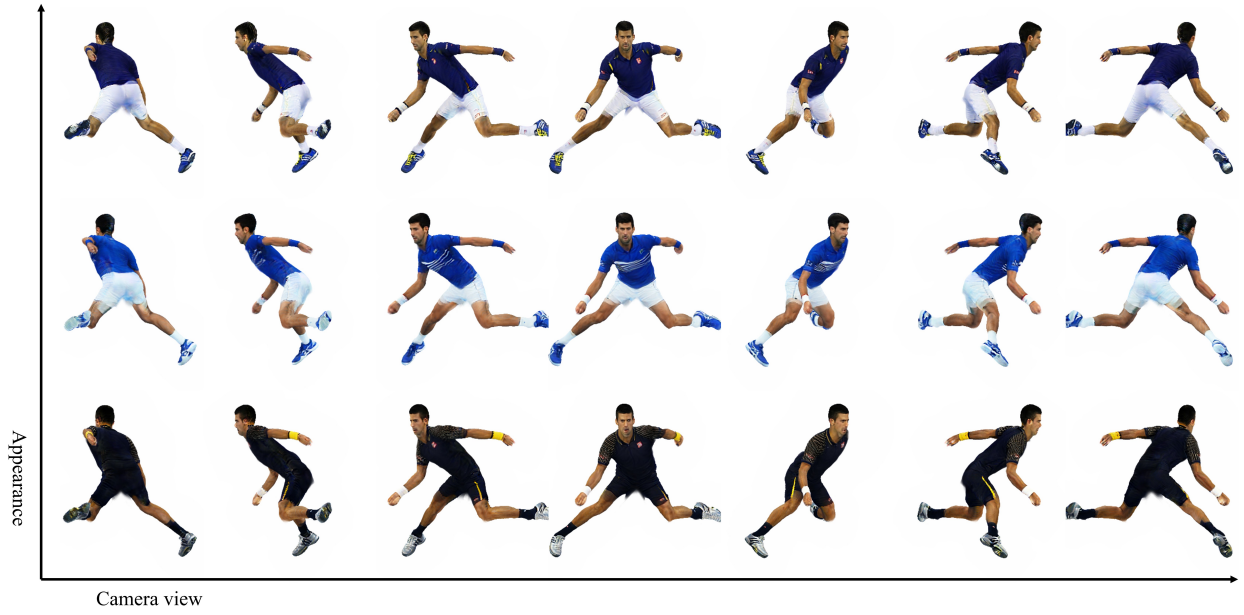


Figure 5.10: Visualization of the (appearance, camera view) plane of the reconstructed space of Novak Djokovic.

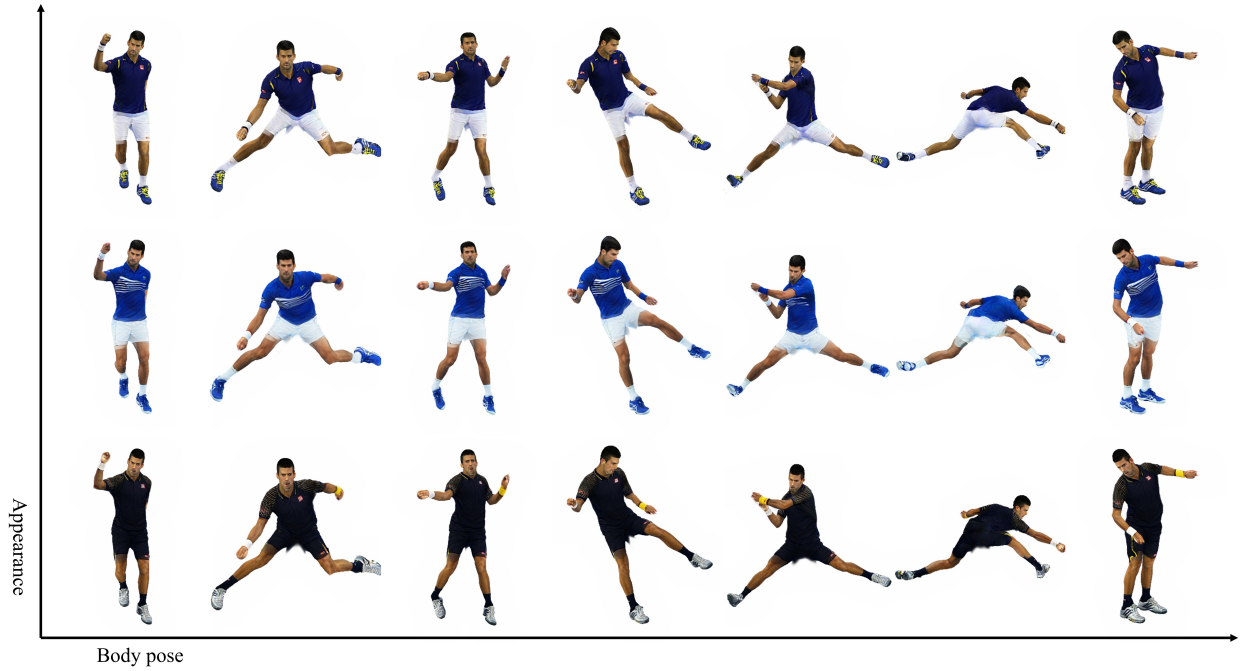


Figure 5.11: Visualization of the (appearance, body pose) plane of the reconstructed space of Novak Djokovic.

Figure 5.12: Visualization of the (body pose, camera view) plane of the reconstructed space of Novak Djokovic.

Figure 5.13: Visualization of the (appearance, camera view) plane of the reconstructed space of Serena Willaims.

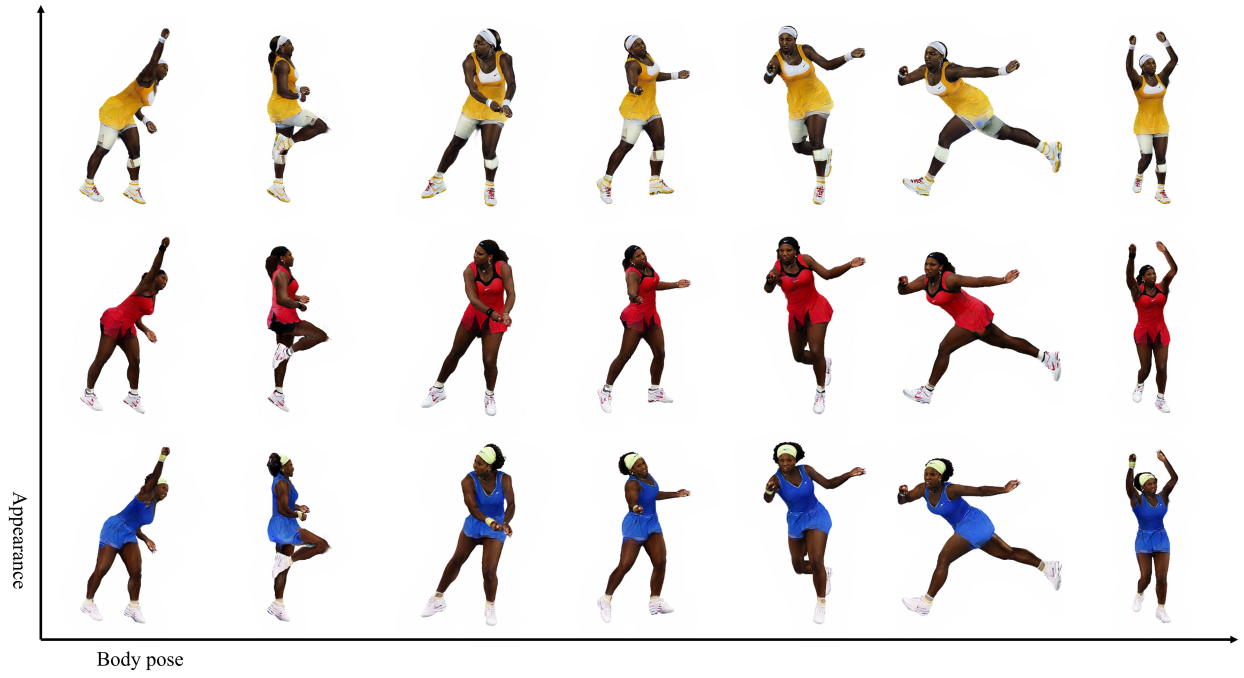


Figure 5.14: Visualization of the (appearance, body pose) plane of the reconstructed space of Serena Williams.

Figure 5.15: Visualization of the (body pose, camera view) plane of the reconstructed space of Serena Willaims.

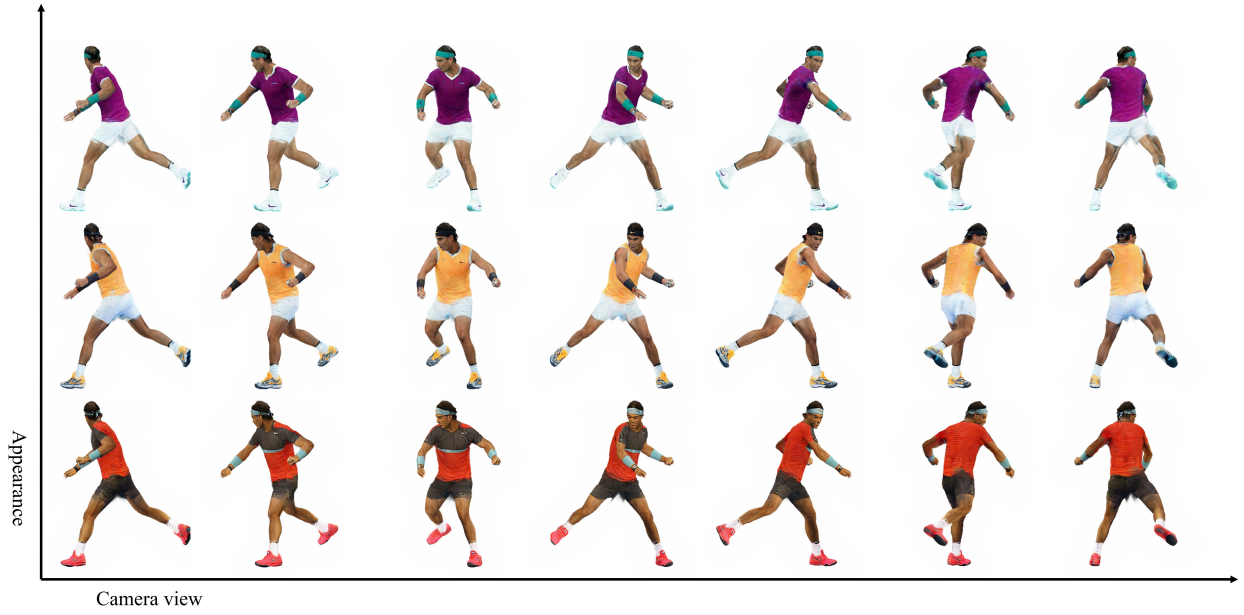


Figure 5.16: Visualization of the (appearance, camera view) plane of the reconstructed space of Rafael Nadal.

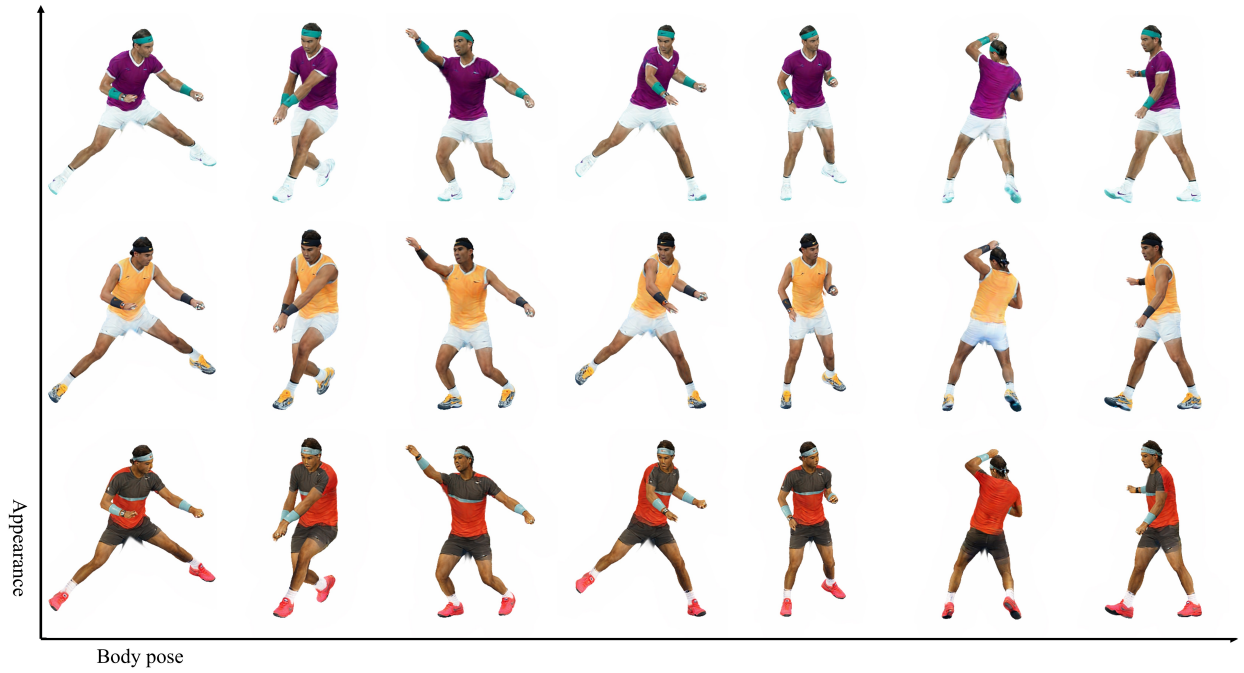


Figure 5.17: Visualization of the (appearance, body pose) plane of the reconstructed space of Rafael Nadal.

Figure 5.18: Visualization of the (body pose, camera view) plane of the reconstructed space of Rafael Nadal.

Chapter 6

DISCUSSIONS AND CONCLUSION

In this chapter, I wrap up this thesis by sharing my thoughts on storytelling (Sec. 6.1), the process of selecting research problems (Sec. 6.2), and the societal impact of synthesized contents (Sec. 6.3). Following that, I discuss potential directions for future research (Sec. 6.4), and finally, I conclude the thesis with my reflections. (Sec. 6.5).

6.1 Story First

Essentially, the "story first" principle is all about reminding storytellers and technicians such as computer scientists to keep their eyes on the main goal: telling a good story. In this thesis, I show how to introduce new dimensions such as space and time into visual storytelling. However, the decision of whether and how to utilize these techniques will ultimately depend on the specific requirements and demands of the narrative.

A good example is the movie "The Matrix." In its most iconic scene, the incorporation of free-viewpoint rendering, commonly known as the bullet-time effect, goes beyond its visually stunning moments. It serves to visually depict the characters' power to manipulate space and time, effectively highlighting the contrast between the real world and the simulated reality. This effect not only reinforces the film's themes of challenging what is reality but also pushes the limits of human perception.

It's very likely that within the next 3 to 5 years, the approaches I've proposed for human rendering and reconstruction could be replaced by even more advanced techniques that offer significantly better quality, especially given the rapid advancement of artificial intelligence. However, having access to the most advanced technology doesn't necessarily guarantee the ability to tell a better story. Telling a good story relies heavily on close collaboration between storytellers, artists, and

computer scientists.

6.2 Research Problem Selection

Regarding the development of technology, there are generally two types of strategies: the bottom-up approach and the top-down approach. The bottom-up approach involves examining the existing technology available to us and building upon it to discover new applications and capabilities. In contrast, the top-down approach focuses on developing technologies specifically tailored to meet the requirements of telling a compelling story. Both strategies hold equal significance: the former aims to push the boundaries of technology, while the latter ensures the delivery of a convincing narrative.

The same principle also applies to the selection of research problems. One can either focus on enhancing existing methods or approach the problem with a specific application in mind, figuring out the steps necessary to achieve that particular goal. In my thesis, I was primarily motivated by applications and developing related approaches. However, throughout the process, the bottom-up approach plays a crucial role as we must first improve the existing methods to determine their suitability for tailored application towards a specific goal. The line separating these two approaches turns out to be unclear. However, the fact itself reconfirms that both strategies hold equal importance in achieving substantial progress in either the field of science or art.

6.3 Synthesis and Reality

The rapid advancement of technology in image, video, and 3D synthesis blurs the line between synthesized reality and actual facts. It has a dual impact. On one hand, it offers tremendous potential to enhance storytelling, elevate visual experiences, and enable seamless remote communications. These advancements can contribute to more immersive narratives, improved visual effects in movies, and bridge the gap between people across distances.

On the other hand, there are concerns regarding the misuse of such technology. It opens the door for the creation and dissemination of fake messages, offensive content, and the potential violation of personal privacy. The ability to manipulate media with advanced AI technologies raises

ethical, legal, and social challenges. Deepfakes, for example, can be used to deceive, misinform, or damage reputations, leading to significant consequences in personal and public domains.

Indeed, the issue of blurred reality has been present for quite some time, because of the widespread use of digital visual effects in films and the availability of powerful video and photo editing tools such as Adobe Photoshop. However, its significance has grown exponentially with the advance of cutting-edge technologies such as diffusion models [131, 59, 129]. Striking a balance between creative and positive applications while addressing the risks is challenging but essential. Addressing the problem requires a multidisciplinary approach involving technology, policy, education, and public awareness.

6.4 Future Works

Reconstructing in full detail My proposed approaches on human reconstruction have primarily concentrated on modeling body movement while disregarding facial expressions or hand motions. With the help of existing head [87], hand [133], or even full-body parametric models such as SMPL-X [123], proposing a pipeline that effectively captures an individual that includes all of these subtle details in the application of free-viewpoint rendering represents a promising direction.

Generating photorealistic images The hallmark of rendering is producing photorealistic images. While current results are high-quality, there remains much room for further improvement. This could entail recovering high-detail surfaces, reconstructing environmental lighting, and estimating material properties (i.e., BRDF functions) from sparse or even single viewpoints. Recently, diffusion models have demonstrated success in generating images [131, 135, 132], videos [59, 150], and 3D models [129, 91], with realistic details, offering a promising alternative to addressing the issue.

Casual capture Cao et al. [17] have showcased a fascinating technique for reconstructing a 3D head avatar using photos taken with mobile phones. While the results are encouraging, creating a full-body avatar, not only the head, from casual captures remains difficult. HumanNeRF and

PersonNeRF, presented in this thesis, demonstrate promising approaches toward this goal, with HumanNeRF focusing on videos and PersonNeRF centered around photo collections as their respective inputs. Another possible solution could involve integrating a generative model for human avatars, trained on a wide range of data including both realistic and synthetic images. A personalized avatar can be generated by fine-tuning this model with personal photo scans. Exploring how to effectively combine images, videos, and synthetic data to develop such a powerful generative model presents an exciting problem that warrants further investigation.

6.5 Conclusion

In my thesis, I have explored the problem of reconstructing and rendering humans from unstructured data like internet images or YouTube videos. The primary goal is to enhance the visual storytelling experience by introducing new dimensions such as time and space. I have reviewed the literature in this domain and have proposed three approaches that lead to different applications: Photo Wake-Up [176] allows for creating 3D human animations on AR devices like HoloLens using only single images; HumanNeRF [178] enables free-viewpoint rendering of moving persons from a YouTube video; PersonNeRF [179] is capable of reconstructing a subject such as Roger Federer from internet photo collections, enabling three-dimensional rendering and time-lapse animations. These advancements push the boundaries of technology for storytelling and remote communication while making it accessible to a wider audience beyond professional studios.

As computer scientists, we are often fascinated by those exciting "Aha!" moments when everything suddenly works and we are always amazed by the rapid progress in artificial intelligence, which consistently delivers astonishing results. However, it is equally crucial for us to acknowledge the potential risks and the possible misuse of this powerful tool. Just like telling a fascinating story relies on the dynamic collaboration among storytellers, artists, and computer scientists, building a brighter future requires the collective effort of everyone involved. Technology alone is not sufficient to create a compelling narrative, and the same principle holds true for constructing a better world as well. After all, we all aspire to tell a beautiful story for ourselves, don't we?

BIBLIOGRAPHY

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 63

[2] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *2018 International Conference on 3D Vision (3DV)*, pages 98–109. IEEE, 2018. 21

[3] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8387–8397, 2018. 12, 20, 21, 23, 33, 34, 115

[4] Thiemo Alldieck, Mihai Zanfir, and Cristian Sminchisescu. Photorealistic monocular 3D reconstruction of humans wearing clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 17

[5] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)*, 22(3):587–594, 2003. 12

[6] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. *ACM SIGGRAPH 2005*, 2005. 12

[7] Hadar Averbuch-Elor, Daniel Cohen-Or, Johannes Kopf, and Michael F. Cohen. Bringing portraits to life. *ACM Transactions on Graphics (Proceeding of SIGGRAPH Asia 2017)*, 36(4):to appear, 2017. 4, 5, 20

[8] Jiamin Bai, Aseem Agarwala, Maneesh Agrawala, and Ravi Ramamoorthi. Automatic cinemagraph portraits. In *Computer Graphics Forum*, volume 32, pages 17–25. Wiley Online Library, 2013. 20

[9] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8340–8348, 2018. 5, 14, 42

[10] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24–1, 2009. 21, 31, 111

[11] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021. 40, 41, 48

[12] Ronen Basri, David Jacobs, and Ira Kemelmacher. Photometric stereo with general, unknown lighting. *International Journal of Computer Vision*, 72(3):239–257, 2007. 24

[13] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. LoopReg: Self-supervised learning of implicit surface correspondences, pose and shape for 3D human mesh registration. *Advances in Neural Information Processing Systems*, 33, 2020. 45

[14] 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. *ECCV*, 2016. 12, 21, 23, 24, 33, 34

[15] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017. 68

[16] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001. 29, 30

[17] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, et al. Authentic volumetric avatars from a phone scan. *ACM Transactions on Graphics (TOG)*, 41(4):1–19, 2022. 90

[18] Joel Carranza, Christian Theobalt, Marcus A Magnor, and Hans-Peter Seidel. Free-viewpoint video of human actors. *ACM transactions on graphics (TOG)*, 22(3):569–577, 2003. 8, 11, 40, 41, 63

[19] Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4D video textures for interactive character appearance. *Computer Graphics Forum*, 2014. 40, 41

[20] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5933–5942, 2019. 14, 42

[21] Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)*, 2013. 41

[22] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. *SIGGRAPH*, 1993. 41

[23] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. SNARF: Differentiable forward skinning for animating non-rigid neural implicit shapes. *ICCV*, 2021. 45, 63

[24] Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H Salesin, and Richard Szeliski. Animating pictures with stochastic motion textures. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 853–860. ACM, 2005. 4, 20

[25] CMU. *CMU Graphics Lab Motion Capture Database*, 2007. 32

[26] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 2015. 13, 41, 63

[27] Nikkatsu Corp. http://intl.nikkatsu.com/. 32

[28] Apple Corps. 35, 36, 37

[29] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 63

[30] Dark-Crawler. https://www.deviantart.com/dark-crawler/art/Son-Goku-465455963. 34, 36

[31] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM SIGGRAPH 2008*, 2008. 12, 41

[32] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. *SIGGRAPH*, 1996. 41, 63

[33] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. *ECCV*, 2020. 45

[34] Endri Dibra, Himanshu Jain, Cengiz Oztireli, Remo Ziegler, and Markus Gross. Human shape from silhouettes using generative HKS descriptors and cross-modal neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4826–4836, 2017. 21

[35] Zijian Dong, Chen Guo, Jie Song, Xu Chen, Andreas Geiger, and Otmar Hilliges. PINA: Learning a personalized implicit neural avatar from a single RGB-D video sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20470–20480, 2022. 64

[36] 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. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. 13, 63

[37] WikiArt: Visual Art Encyclopedia. https://www.wikiart.org/. 32, 34, 36

[38] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational U-Net for conditional appearance and shape generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8857–8866, 2018. 31, 37

[39] Zeev Farbman, Gil Hoffer, Yaron Lipman, Daniel Cohen-Or, and Dani Lischinski. Coordinates for instant image cloning. In *ACM Transactions on Graphics (TOG)*, volume 28, page 67. ACM, 2009. 111

[40] Matthew Flagg, Atsushi Nakazawa, Qiushuang Zhang, Sing Bing Kang, Young Kee Ryu, Irfan Essa, and James M Rehg. Human video textures. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 199–206, 2009. 20, 40

[41] Michael S Floater. Mean value coordinates. *Computer aided geometric design*, 20(1):19–27, 2003. 25

[42] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 63

[43] Juergen Gall, Carsten Stoll, Edilson De Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel. Motion capture using joint skeleton tracking and surface estimation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1753. Ieee, 2009. 12

[44] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. *ICCV*, 2021. 41

[45] GettyImages. https://www.gettyimages.com/. 32, 36

[46] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 8, 9, 41

[47] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018. 21

[48] 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. *ACM Transactions on Graphics (ToG)*, 38(6):1–19, 2019. 13, 40, 41, 63

[49] Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. Real-time deep dynamic characters. *ACM Transactions on Graphics (TOG)*, 2021. 42

[50] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. LiveCap: Real-time human performance capture from monocular video. *ACM Transactions On Graphics (TOG)*, 38(2):1–17, 2019. 12

[51] 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*, pages 5052–5063, 2020. 12

[52] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 21

[53] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. ARCH++: Animation-ready clothed human reconstruction revisited. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11046–11056, 2021. 17, 42

[54] Peter Hedman and Johannes Kopf. Instant 3D photography. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018. 41

[55] Peter Hedman, Tobias Ritschel, George Drettakis, and Gabriel Brostow. Scalable inside-out image-based rendering. *ACM Transactions on Graphics (TOG)*, 2016. 41

[56] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021. 41

[57] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. SAPE: Spatially-adaptive progressive encoding for neural optimization. *Advances in Neural Information Processing Systems*, 34:8820–8832, 2021. 48

[58] 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. *Advances in neural information processing systems*, 30, 2017. 71

[59] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 90

[60] Aleksander Holynski, Brian L. Curless, Steven M. Seitz, and Richard Szeliski. Animating pictures with eulerian motion fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5810–5819, June 2021. 4

[61] Alexander Hornung, Ellen Dekkers, and Leif Kobbelt. Character animation from 2D pictures and 3d motion data. *ACM Transactions on Graphics (TOG)*, 26(1):1, 2007. 5, 20, 21, 34, 35

[62] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. ARCH: Animatable reconstruction of clothed humans. *CVPR*, 2020. 17, 42, 45

[63] Intel. *Intel®True View*, 2017. 8

[64] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 8, 14

[65] Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. MovieReshape: Tracking and reshaping of humans in videos. In *ACM Transactions on Graphics (TOG)*, volume 29, page 148. ACM, 2010. 21

[66] Zhang Jiakai, Liu Xinhang, Ye Xinyi, Zhao Fuqiang, Zhang Yanshun, Wu Minye, Zhang Yingliang, Xu Lan, and Yu Jingyi. Editable free-viewpoint video using a layered neural representation. *ACM SIGGRAPH*, 2021. 42

[67] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 402–418. Springer, 2022. 64

[68] Neel Joshi, Sisil Mehta, Steven Drucker, Eric Stollnitz, Hugues Hoppe, Matt Uyttendaele, and Michael Cohen. Cliplets: juxtaposing still and dynamic imagery. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 251–260. ACM, 2012. 20

[69] Takeo Kanade, Peter Rander, and PJ Narayanan. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia*, 4(1):34–47, 1997. 8, 9, 40, 41, 63

[70] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018. 12, 21

[71] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 21

[72] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5614–5623, 2019. 12

[73] Ira Kemelmacher-Shlizerman. Internet based morphable model. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013. 63

[74] Ira Kemelmacher-Shlizerman and Steven M Seitz. Face reconstruction in the wild. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1746–1753. IEEE, 2011. 31, 109

[75] Ira Kemelmacher-Shlizerman, Eli Shechtman, Rahul Garg, and Steven M Seitz. Exploring photobios. In *ACM Transactions on Graphics (TOG)*, volume 30, page 61. ACM, 2011. 5, 6, 20

[76] Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 3D object manipulation in a single photograph using stock 3d models. *ACM Transactions on Graphics (TOG)*, 33(4):127, 2014. 20

[77] Davis E King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10(Jul):1755–1758, 2009. 31, 109

[78] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 50, 71

[79] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020. 12

[80] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. *ICCV*, 2019. 12, 50, 71

[81] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 21

[82] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence*, 16(2):150–162, 1994. 8

[83] Jason Lawrence, Dan B Goldman, Supreeth Achar, Gregory Major Blascovich, Joseph G. Desloge, Tommy Fortes, Eric M. Gomez, Sascha Häberling, Hugues Hoppe, Andy Huibers, Claude Knaus, Brian Kuschak, Ricardo Martin-Brualla, Harris Nover, Andrew Ian Russell, Steven M. Seitz, and Kevin Tong. Project Starline: A high-fidelity telepresence system. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 40(6), 2021. 2, 8

[84] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 8, 9, 41

[85] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. TAVA: Template-free animatable volumetric actors. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 419–436. Springer, 2022. 63

[86] Ruilong Li, Yuliang Xiu, Shunsuke Saito, Zeng Huang, Kyle Olszewski, and Hao Li. Monocular real-time volumetric performance capture. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 49–67. Springer, 2020. 17

[87] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM SIGGRAPH*, 2017. 90

[88] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *CVPR*, 2021. 41

[89] Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. Crowdsampling the plenoptic function. In *European Conference on Computer Vision*, pages 178–196. Springer, 2020. 63

[90] Shu Liang, Linda G Shapiro, and Ira Kemelmacher-Shlizerman. Head reconstruction from internet photos. In *European Conference on Computer Vision*, pages 360–374. Springer, 2016. 63

[91] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3d content creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 90

[92] 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. *ACM Trans. Graph.(ACM SIGGRAPH Asia)*, 2021. 42, 63

[93] Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhöfer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, and Christian Theobalt. Neural human video rendering by learning dynamic textures and rendering-to-video translation. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 40, 42

[94] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural Volumes: Learning dynamic renderable volumes from images. *ACM SIGGRAPH*, 2019. 16, 68

[95] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 9, 12, 17, 23, 40, 42

[96] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3D shape reconstruction from sketches via multi-view convolutional networks. In *3D Vision (3DV), 2017 International Conference on*, pages 67–77. IEEE, 2017. 21

[97] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. *Advances in neural information processing systems*, 30, 2017. 14, 31, 37, 42

[98] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 99–108, 2018. 31, 37

[99] Qianli Ma, Shunsuke Saito, Jinlong Yang, Siyu Tang, and Michael J Black. SCALE: Modeling clothed humans with a surface codec of articulated local elements. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16082–16093, 2021. 63

[100] Ricardo Martin-Brualla, David Gallup, and Steven M Seitz. 3D time-lapse reconstruction from internet photos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1332–1340, 2015. 63

[101] Ricardo Martin-Brualla, David Gallup, and Steven M Seitz. Time-lapse mining from internet photos. *ACM Transactions on Graphics (TOG)*, 34(4):62, 2015. 5, 20, 63

[102] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, et al. LookinGood: Enhancing performance capture with real-time neural re-rendering. *ACM Transactions on Graphics (TOG)*, 2018. 14, 40, 42, 63

[103] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *CVPR*, 2021. 40, 63, 68

[104] Matterport. *Mask R-CNN Implementation by Matterport, Inc*, 2017. 21

[105] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J Gortler, and Leonard McMillan. Image-based visual hulls. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 369–374, 2000. 40, 41, 63

[106] Kevin Matzen and Noah Snavely. Scene chronology. In *European conference on computer vision*, pages 615–630. Springer, 2014. 63

[107] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1995. 47

[108] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019. 14, 63

[109] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. LEAP: Learning articulated occupancy of people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10461–10471, 2021. 45, 63

[110] 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. *ECCV*, 2020. 16, 17, 41, 44, 47, 48, 63, 65, 113

[111] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 60

[112] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. *ECCV*, 2018. 42

[113] Richard A Newcombe, Dieter Fox, and Steven M Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015. 13

[114] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 67

[115] Michael Niemeyer and Andreas Geiger. GIRAFFE: Representing scenes as compositional generative neural feature fields. *CVPR*, 2021. 41

[116] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. *ICCV*, 2021. 42

[117] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 International Conference on 3D Vision (3DV)*, pages 484–494. IEEE, 2018. 21

[118] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. Holoportation: Virtual 3D teleportation in real-time. In *Proceedings of the 29th annual symposium on user interface software and technology*, pages 741–754, 2016. 8

[119] Rohit Pandey, Anastasia Tkach, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Ricardo Martin-Brualla, Andrea Tagliasacchi, George Papandreou, Philip Davidson, Cem Keskin, et al. Volumetric capture of humans with a single RGBD camera via semi-parametric learning. *CVPR*, 2019. 42

[120] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 17

[121] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. 40, 41, 48

[122] 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. 40, 41, 48, 51

[123] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 90

[124] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 459–468, 2018. 21

[125] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. *ICCV*, 2021. 40, 42, 45

[126] 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. *CVPR*, 2021. 15, 40, 42, 50, 51, 52, 55, 58, 63, 118

[127] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM Transactions on graphics (TOG)*, volume 22, pages 313–318. ACM, 2003. 31

[128] Kerry Varnum Photography. https://www.facebook.com/KerryVarnumPhotography. 32, 37

[129] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 90

[130] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. *CVPR*, 2020. 40, 41

[131] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 90

[132] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 90

[133] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM SIGGRAPH Asia*, 2017. 90

[134] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004. 30

[135] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 90

[136] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitiza-

tion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2304–2314, 2019. 17

[137] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In *CVPR*, 2020. 17

[138] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J Black. SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2886–2897, 2021. 45, 63

[139] Soubhik Sanyal, Alex Vorobiov, Timo Bolkart, Matthew Loper, Betty Mohler, Larry S Davis, Javier Romero, and Michael J Black. Learning realistic human reposing using cyclic self-supervision with 3D shape, pose, and appearance consistency. In *CVPR*, 2021. 42

[140] Kripasindhu Sarkar, Vladislav Golyanik, Lingjie Liu, and Christian Theobalt. Style and pose control for image synthesis of humans from a single monocular view. *arXiv preprint arXiv:2102.11263*, 2021. 42

[141] Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. In *ACM transactions on graphics (TOG)*, volume 25, pages 533–540. ACM, 2006. 110

[142] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47:7–42, 2002. 8

[143] Arno Schödl, Richard Szeliski, David H Salesin, and Irfan Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498. ACM Press/Addison-Wesley Publishing Co., 2000. 20

[144] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 63

[145] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. *ECCV*, 2016. 42

[146] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3d-aware image synthesis. *NeurIPS*, 2020. 48

[147] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 519–528. IEEE, 2006. 63

[148] Qi Shan, Riley Adams, Brian Curless, Yasutaka Furukawa, and Steven M Seitz. The visual turing test for scene reconstruction. In *2013 International Conference on 3D Vision-3DV 2013*, pages 25–32. IEEE, 2013. 63

[149] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. *Visual Communications and Image Processing 2000*, 2000. 41

[150] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 90

[151] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo Tourism: exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, 2006. 63

[152] Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. NeRV: Neural reflectance and visibility fields for relighting and view synthesis. *CVPR*, 2021. 41

[153] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE computer graphics and applications*, 27(3):21–31, 2007. 8, 10

[154] Jonathan Starck, Gregor Miller, and Adrian Hilton. Video-based character animation. *ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2005. 40, 41

[155] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-NeRF: Articulated neural radiance fields for learning human shape, appearance, and pose. In *Advances in Neural Information Processing Systems*, 2021. 42

[156] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural 3D reconstruction in the wild. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 63

[157] Superherohype.com. 32, 36

[158] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022. 41, 63

[159] Tabby. http://tabbythis.com/?p=1326. 35

[160] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 63

[161] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 41, 48

[162] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. *Computer Graphics Forum*, 2020. 8, 63

[163] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. *Computer Graphics Forum*, 2022. 8, 63

[164] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 2019. 15

[165] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-GIF: Neural generalized implicit functions for animating people in clothing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11708–11718, 2021. 45, 63

[166] James Tompkin, Fabrizio Pece, Kartic Subr, and Jan Kautz. Towards moment imagery: Automatic cinemagraphs. In *Visual Media Production (CVMP), 2011 Conference for*, pages 87–93. IEEE, 2011. 20

[167] David Traum, Andrew Jones, Kia Hays, Heather Maio, Oleg Alexander, Ron Artstein, Paul Debevec, Alesia Gainer, Kallirroi Georgila, Kathleen Haase, et al. New dimensions in testimony: Digitally preserving a holocaust survivor's interactive storytelling. In *Interactive Storytelling: 8th International Conference on Interactive Digital Storytelling, ICIDS 2015, Copenhagen, Denmark, November 30-December 4, 2015, Proceedings 8*, pages 269–281. Springer, 2015. 2

[168] 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 dynamic scene from monocular video. *ICCV*, 2021. 40, 41

[169] Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *ECCV*, 2018. 21

[170] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 21

[171] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. *ACM SIGGRAPH 2008*, 2008. 8, 11, 12, 40, 41

[172] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning multi-view image-based rendering. *CVPR*, 2021. 40

[173] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *NeurIPS*, 2018. 14, 42

[174] Tuanfeng Y Wang, Duygu Ceylan, Krishna Kumar Singh, and Niloy J Mitra. Dance in the wild: Monocular human animation with neural dynamic appearance synthesis. In *2021 International Conference on 3D Vision (3DV)*, pages 268–277. IEEE, 2021. 42

[175] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 21

[176] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo Wake-Up: 3D character animation from a single photo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5908–5917, 2019. 5, 17, 18, 91

[177] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Vid2Actor: Free-viewpoint animatable person synthesis from video in the wild. *arXiv preprint arXiv:2012.12884*, 2020. 42, 44, 45

[178] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16210–16220, June 2022. 3, 17, 39, 62, 64, 66, 71, 72, 73, 74, 78, 79, 91

[179] Chung-Yi Weng, Pratul P. Srinivasan, Brian Curless, and Ira Kemelmacher-Shlizerman. PersonNeRF: Personalized reconstruction from photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 524–533, June 2023. 6, 17, 62, 91

[180] Wikimedia. https://commons.wikimedia.org/wiki/Main_Page. 36, 37

[181] Wikipedia. *Bullet Time*. 2, 8

[182] Minye Wu, Yuehao Wang, Qiang Hu, and Jingyi Yu. Multi-view neural human rendering. *CVPR*, 2020. 42

[183] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. *CVPR*, 2021. 41

[184] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022. 63

[185] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. ICON: Implicit Clothed humans Obtained from Normals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13296–13306, June 2022. 17

[186] Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. Video-based characters: creating new human performances from a multi-view video database. *ACM Transactions on Graphics (TOG)*, 30(4):32, 2011. 20, 41

[187] Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-NeRF: Neural radiance fields for rendering and temporal reconstruction of humans in motion. *Advances in Neural Information Processing Systems*, 34, 2021. 42

[188] Weipeng Xu, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt. MonoPerfCap: Human performance capture from monocular video. *ACM Transactions on Graphics (ToG)*, 37(2):1–15, 2018. 12

[189] Xuemiao Xu, Liang Wan, Xiaopei Liu, Tien-Tsin Wong, Liansheng Wang, and Chi-Sing Leung. Animating animal motion from still. In *ACM Transactions on Graphics (TOG)*, volume 27, page 117. ACM, 2008. 5, 20

[190] Ze Yang, Shenlong Wang, Siva Manivasagam, Zeng Huang, Wei-Chiu Ma, Xinchen Yan, Ersin Yumer, and Raquel Urtasun. S3: Neural shape, skeleton, and skinning fields for 3d human modeling. *CVPR*, 2021. 42, 45

[191] 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 *The IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, June 2018. 13

[192] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. NeRF++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 40, 41

[193] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*, 2018. 48, 52, 65

[194] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. NeRFactor: Neural factorization of shape and reflectance under an unknown illumination. *SIGGRAPH Asia*, 2021. 41

[195] Zerong Zheng, Tao Yu, Yebin Liu, and Qionghai Dai. PAMIR: Parametric model-conditioned implicit representation for image-based human reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):3170–3184, 2021. 17

[196] Shizhe Zhou, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han. Parametric reshaping of human bodies in images. In *ACM Transactions on Graphics (TOG)*, volume 29, page 126. ACM, 2010. 21

[197] Zhenglong Zhou, Bo Shu, Shaojie Zhuo, Xiaoming Deng, Ping Tan, and Stephen Lin. Image-based clothes animation for virtual fitting. In *SIGGRAPH Asia 2012 Technical Briefs*, page 33. ACM, 2012. 20

[198] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3):600–608, 2004. 8, 10, 41, 63

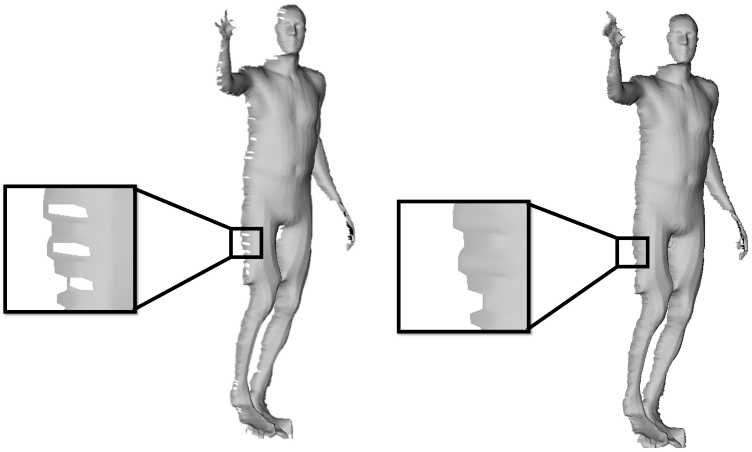
[199] Silvia Zuffi, Angjoo Kanazawa, and Michael J. Black. Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018. 21

Appendix A

PHOTO WAKE-UP ADDITIONAL DETAILS

A.1 Mesh Hole-filling

In practice, holes may arise when warping by $f(x)$, i.e., small regions in which $f(x) \notin S_{\text{SMPL}}$. An example mesh rebuilt from an ouput depth map with holes is illustrated in Fig. A.1(a). To fill these holes, we again apply mean value coordinates, now for smooth inpainting of normals and skinning weights. For each hole H in normal map N, we collect the points on the hole boundary ∂H, compute mean-value coordinates of the points $x \in H$ in terms of the boundary points, and then interpolate the boundary normals using these coordinates. We do the same for the skinning map W, interpolating the vector skinning weights around ∂H. A mesh result before and after hole-filling is shown in Fig. A.1.



(a) w/o hole filling (b) with hole filling

Figure A.1: Hole-filling results. Holes result can arise when warping between silhouettes. Here we visualize the result of holes in the depth map (a), which we then smoothly fill (b).

(a) Input image (b) Before correction (c) After correction

Figure A.2: In the result of SMPL optimization the head pose is usually incorrect (b), thus we further align the head to detected face fiducials and estimate head pose. The correction result is shown in (c).

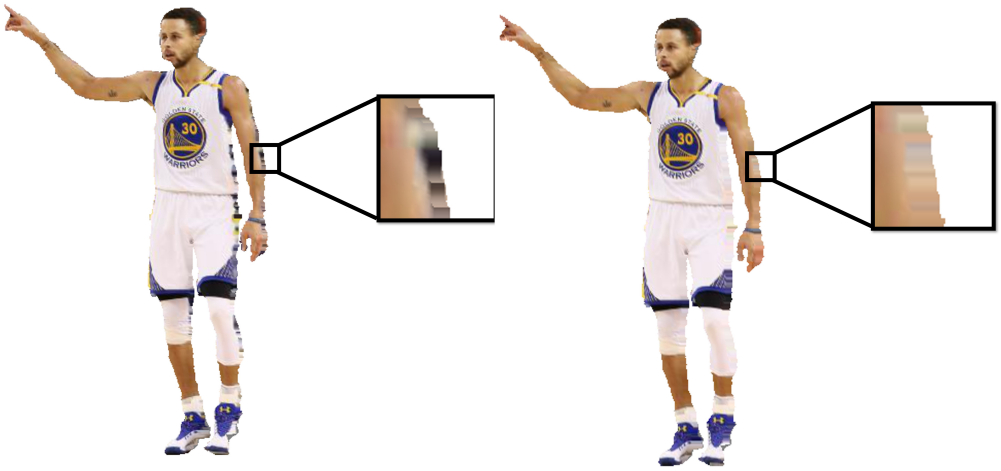
A.2 Facial Region Alignment

The generic body shape model includes a head model which is fitted during the process just like the body, but typically the optimization fails to predict correct head pose. Incorrect head pose creates strong artifacts when the shape is textured and viewed from the sides. We found aligning the head with face-specific techniques improves our animation results significantly. We begin by estimating the head pose and then warp the transformed head mesh according to the image, as explained as follows.

A head pose is estimated as in [74]. Specifically, 2D fiducial points q are first detected using [77] on the photo. Then predefined 3D fiducial points Q on the SMPL mesh are used to recover the head pose by solving $q = sRQ + T$, where s is a scale value, R is a rotation matrix, and T is a translation vector. To recover R, we first subtract the centroid from both point sets to get $p = q - \bar{q}$ and $P = Q - \bar{Q}$, estimate a 2×3 linear transformation A such that $p = AP$, let the third row of A as the cross product of the first two rows to get A', and apply SVD, $A' = UDV^T$, to estimate R as UV^T. An example result after head correction is shown in Fig. A.2(c).

Given the corrected head pose, we generate the full depth map Z as before. We warp Z in the head part to match the fiducials to ensure good texturing. We treat 7 fiducial points F (corners of two eyes, nose, and corners of mouth) in the image and boundary points ∂B of the head as

anchors. We then project the corresponding 3D fiducial points Q on the SMPL mesh into the depth map Z_{SMPL} and apply mean-value coordinates to find their corresponding locations, F', in the output. Finally, we warp the depth map Z within the head part to map F' to F while keeping the boundary ∂B fixed, via moving least squares transform [141]. We chose MLS due to its good global smoothness property, no need for triangulation, and closed-form solution. In case of abstract art or other types of photos where the face and fiducials are not detected, the face-specific treatment process is skipped, and the projected generic mesh is warped to the silhouette just as in the body case.



(a) w/o texture refinement (b) with texture refinement

Figure A.3: (a) When texturing a mesh, errors arise around the silhouette boundary. (b) We reduce the artifacts by replacing them with the colors nearest neighbor pixels well within the silhouette.

A.3 Texturing

Our final step is to texture the reconstructed 3D model. To texture the front mesh (before stitching to the back mesh), we can simply assign colors from the input image to the corresponding vertices back-projected depth map. Due to small errors in person segmentation as well as mixed foreground-background pixels at the silhouette, discolorations may appear on a narrow band near

the boundary of the mesh (Fig. A.3(a)). These errors may be addressable with more sophisticated segmentation refinement and matting. Instead, we simply erode S to form S' and then replace the color of each pixel in $S \setminus S'$ with the color of the nearest pixel in S' (Fig. A.3(b)).

If a frontal region is occluded – e.g., hand in front of the torso – we apply the patch-match algorithm [10] to inpaint the region. Texturing the back of the body is more difficult, as we have no direct observation of it. One approach is to simply mirror the front texture onto the back. This mirrored-texturing produces reasonable results in some cases (e.g., arms), but undesirable results in others (face appears on the back of the head).

To address this problem, we allow the user to choose between mirrored texturing or label-driven texture synthesis – "texture-by-numbers" – on a part-by-part basis. Fig. A.4 illustrates the latter approach. Starting from the original body part label map, the user can apply new color labels to the source (frontal) image, and optionally to the back image. We then synthesize texture for the back, restricted to draw from regions with the same label. When texture synthesis does not produce a satisfactory result, the user can opt instead to revert to mirrored-texturing. Finally, we apply Poisson blending[39] to back texture when stitching it with the front texture.

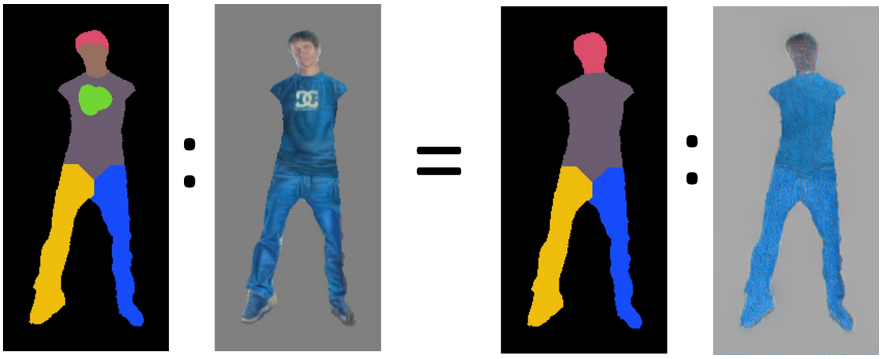


Figure A.4: We transform the back texture construction into a "texture-by-numbers" problem. In the example, we take the estimated body label map to describe both the body front and back. However, for the front part, we roughly paint the regions of the face and the t-shirt logo with new colors, hence excluding the textures of these regions from the body back appearance synthesis.

Appendix B

HUMANNERF ADDITIONAL DETAILS

B.1 Derivation of Motion Bases

We describe how we derive the rotation and translation, $\{R_i, \mathbf{t}_i\}$, to map from bone coordinates in observation space to coordinates in canonical space (Section 3 on "skeletal motion").

We define body pose $\mathbf{p} = (J, \Omega)$, where $J = \{j_i\}$ includes K joint locations and $\Omega = \{\boldsymbol{\omega}_i\}$ defines local joint rotations using axis-angle representations $\in \mathfrak{so}(3)$. Given a predefined canonical pose $\mathbf{p}_c = (J^c, \Omega^c)$ and an observed pose $\mathbf{p} = (J, \Omega)$, the observation-to-canonical transformation M of body part k is:

$$M_k(\mathbf{p}_c, \mathbf{p}) = \prod_{i \in \tau(k)} \begin{bmatrix} \exp(\boldsymbol{\omega}_i^c) & j_i^c \\ 0 & 1 \end{bmatrix} \left\{ \prod_{i \in \tau(k)} \begin{bmatrix} \exp(\boldsymbol{\omega}_i) & j_i \\ 0 & 1 \end{bmatrix} \right\}^{-1}, \tag{B.1}$$

where $\exp(\boldsymbol{\omega}) \in SO(3)$ is a 3×3 rotation matrix computed by taking the exponential of $\boldsymbol{\omega}$ (i.e., applying Rodrigues' rotation formula), and $\tau(k)$ is the ordered set of parents of joint K in the kinematic tree.

The rotation and translation, R_k and t_k, for body part k is can then be extracted from M_k:

$$\begin{bmatrix} R_k & \mathbf{t}_k \\ 0 & 1 \end{bmatrix} = M_k(\mathbf{p}_c, \mathbf{p}). \tag{B.2}$$

B.2 Network Architecture

Figures B.1 - B.4 show the network design for the canonical MLP, the non-rigid motion MLP, the pose correction MLP, and the deep network generating the canonical motion weight volume.

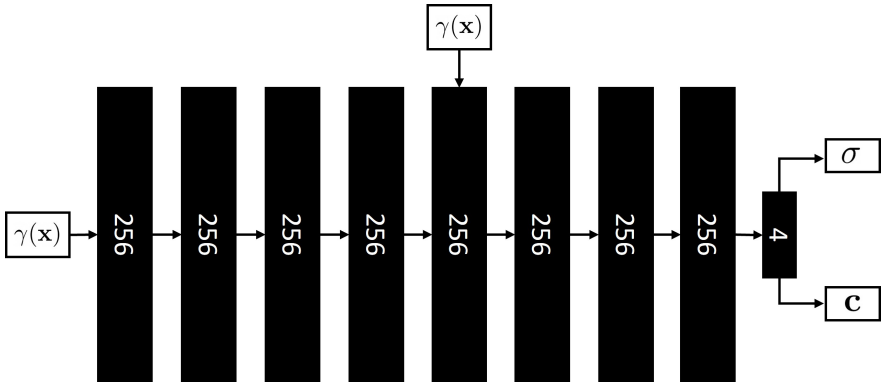


Figure B.1: Canonical MLP visualization. Following NeRF [110], we use an 8-layer MLP with width=256, taking as input positional encoding γ of position $\mathbf{x}$ and producing color $\mathbf{c}$ and density σ. A skip connection that concatenates $\gamma(\mathbf{x})$ to the fifth layer is applied. We adopt ReLU activation after each fully connected layer, except for the one generating color $\mathbf{c}$ where we use *sigmoid*.

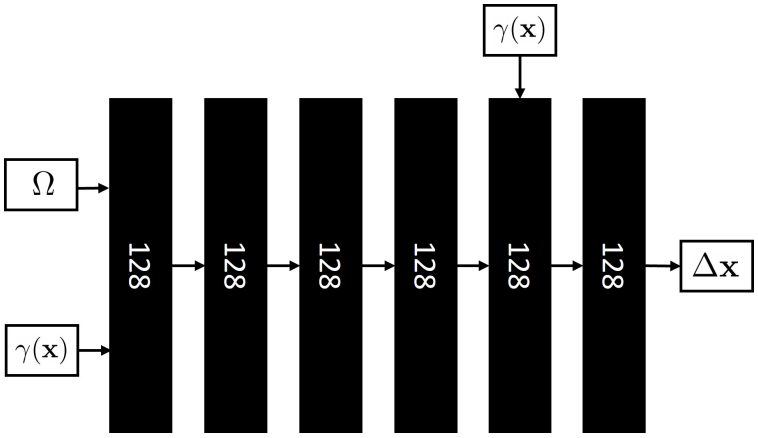


Figure B.2: Non-rigid motion MLP visualization. We choose a 6-layer MLP (width=128) that takes as input the body pose, specifically, joint rotations Ω, and positional encoding, $\gamma(\mathbf{x})$, and predicts the offset $\Delta\mathbf{x}$. We use a skip connection for the positional encoding at the fifth layer. Additionally, we remove the rotation vector of global orientation from joint angles Ω and only uses the remainder as MLP input.

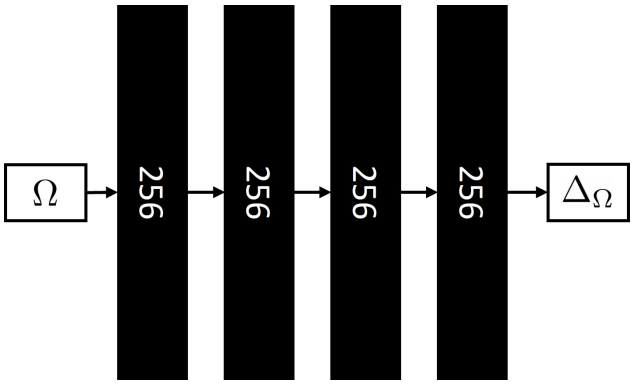


Figure B.3: Pose correction MLP visualization. A 4-layer MLP with width 256 that takes joint angles Ω is used for refining initial poses. Like the non-rigid motion MLP, we take all joints except for root joint (i.e., body orientation) into account and optimize them accordingly.

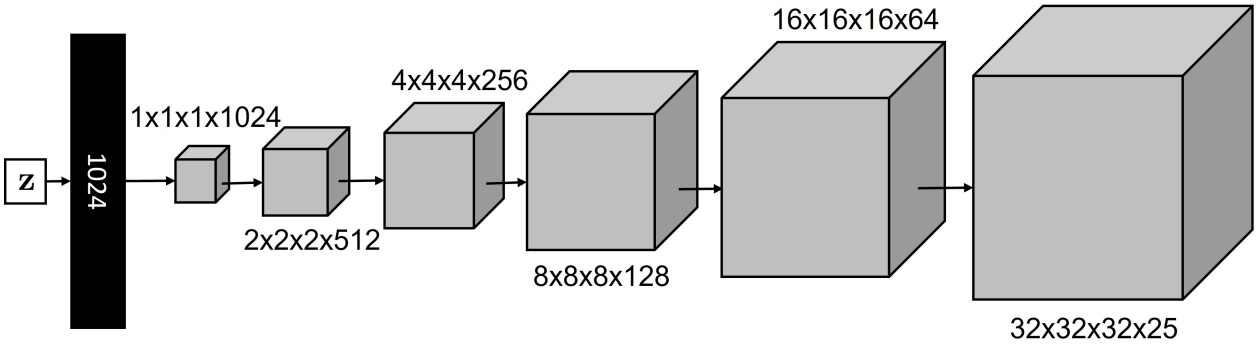
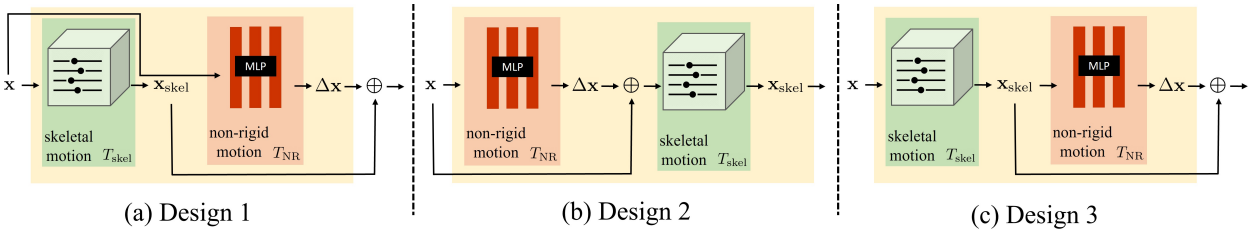


Figure B.4: Network for generating the motion weight volume. The network begins with a fully-connected layer that transforms the (random, constant) latent code $\mathbf{z}$ and reshapes it to a $1 \times 1 \times 1 \times 1024$ grid. Subsequently, it is concatenated with 5 transposed convolutions, increasing volume size while decreasing the number of channels, and finally, produces a volume of size $32 \times 32 \times 32 \times 25$. LeakyReLU is applied after MLP and transposed convolution layers. The size of the latent code $\mathbf{z}$ is 256.



(a) Design 1 (b) Design 2 (c) Design 3

Figure B.5: The three proposed designs of motion decomposition. We choose design 3 (c) as a result of best quality of novel view synthesis, shown in Fig. B.6.

B.3 Motion Field Decomposition

We decompose a motion field into skeletal rigid motion and non-rigid motion. We tested several different formulations for the decomposition. Specifically, starting from a point $\mathbf{x}$ in observation space, we considered three potential decompositions. (To simplify notation and improve readability below, we omit body pose $\mathbf{p}$, which would otherwise always appear as the second argument to each of $T, T_{\text{skel}}, T_{\text{NR}}$.)

(1) Both T_{skel} and T_{NR} conditioned on an observed point position $\mathbf{x}$, illustrated in Fig. B.5-(a):

$$T(\mathbf{x}) = T_{\text{skel}}(\mathbf{x}) + T_{\text{NR}}(\mathbf{x}) \tag{B.3}$$

(2) T_{NR} conditioned on $\mathbf{x}$, but T_{skel} conditioned on position adjusted by non-rigid motion, $\mathbf{x} + T_{\text{NR}}(\mathbf{x})$, illustrated in Fig. B.5-(b):

$$T(\mathbf{x}) = T_{\text{skel}}(\mathbf{x} + T_{\text{NR}}(\mathbf{x})) \tag{B.4}$$

(3) T_{skel} conditioned on $\mathbf{x}$ and T_{NR} conditioned on the position $T_{\text{skel}}(\mathbf{x})$ warped by skeletal rigid motion T_{skel}, illustrated in Fig. B.5-(c):

$$T(\mathbf{x}) = T_{\text{skel}}(\mathbf{x}) + T_{\text{NR}}(T_{\text{skel}}(\mathbf{x})) \tag{B.5}$$

We conducted experiments on the PeopleSnapshot dataset [3], and used 64 samples per ray for

quick evaluation. As shown in Fig. B.6, deforming $\mathbf{x}$ by T_{skel} and then conditioning T_{NR} on that motion (design 3, or Eq. B.5) produces the best quality for novel view synthesis. The result of this experiment explains our final choice of motion decomposition.

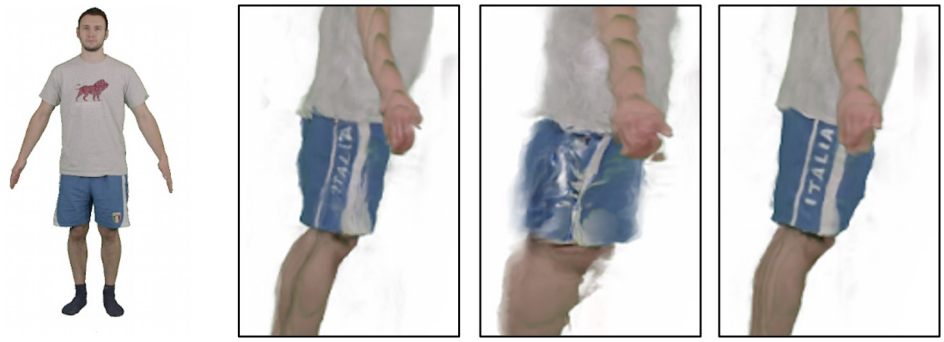


Figure B.6: The experimental result of novel view synthesis on the three proposed motion decompositions, illustrated in Fig. B.5. Design 3 (c) leads to best alignment, the approach we ultimately adopted. In this experiment, we used 64 samples per ray for quick evaluation, introducing color artifacts on the arms not present when using the sampling described in the paper.

Appendix C

PERSONNERF ADDITIONAL DETAILS

C.1 Network Architecture

Fig. C.1 and Fig. C.2 show the network design of our canonical MLP and pose correction MLP. Specifically, we provide the details of how we incorporate appearance embedding ℓ^{app} as well as pose embedding ℓ^{pose} vectors into the corresponding networks.

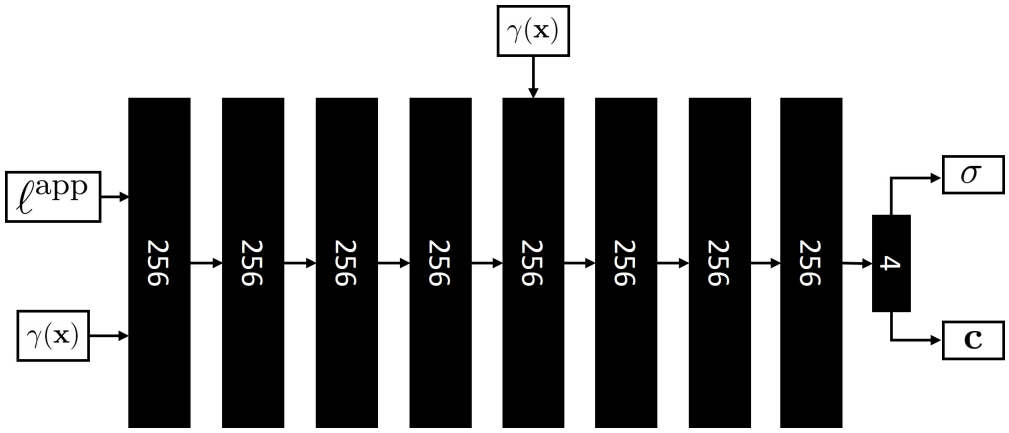


Figure C.1: Canonical MLP netwrok. We use an 8-layer MLP with width=256 that takes as input positional encoding γ of position $\mathbf{x}$ and appearance embedding vector ℓ^{app} with dimension=256. The network outputs color $\mathbf{c}$ and density σ. There is a skip connection that concatenates $\gamma(\mathbf{x})$ to the fifth layer. We use ReLU activations after each fully connected layer. For the output layer, we use a ReLU activation for the density value σ to ensure non-negativity and a *sigmoid* activation for the color $\mathbf{c}$ to constrain values between 0 and 1.

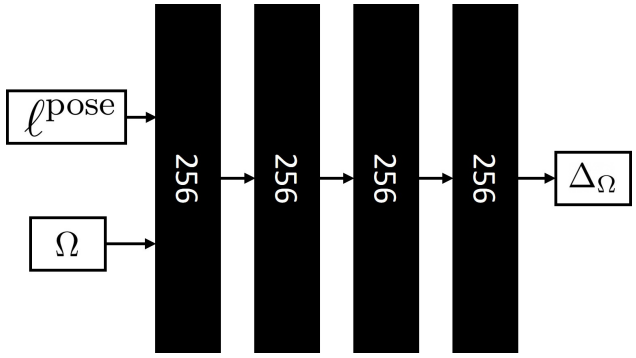


Figure C.2: Pose correction MLP network. We use a 4-layer MLP with width=256 that takes as input joint angles Ω and a pose embedding vector ℓ^{pose} with dimension=16. The network produces the residuals of joint angles that are added back to the input pose to refine the body pose prediction.

C.2 Experiments on ZJU-MoCap dataset

C.2.1 Experimental Setup

We additionally performed experiments on the ZJU-MoCap dataset [126], which provides ground-truth unseen views that enable computation of metrics and analysis of performance on sparse/dense data inputs. We selected subjects 377, 392, and 393–the same individual in different clothing. We evenly selected 10 frames from camera-1 videos to represent "sparse data" (ZJU-Sparse). For "dense data", we used the entire video (ZJU-Dense). The remaining 22 camera views were used for evaluation. We report PSNR, SSIM, and LPIPS* (LPIPS $\times 10^3$) metrics and highlight the best

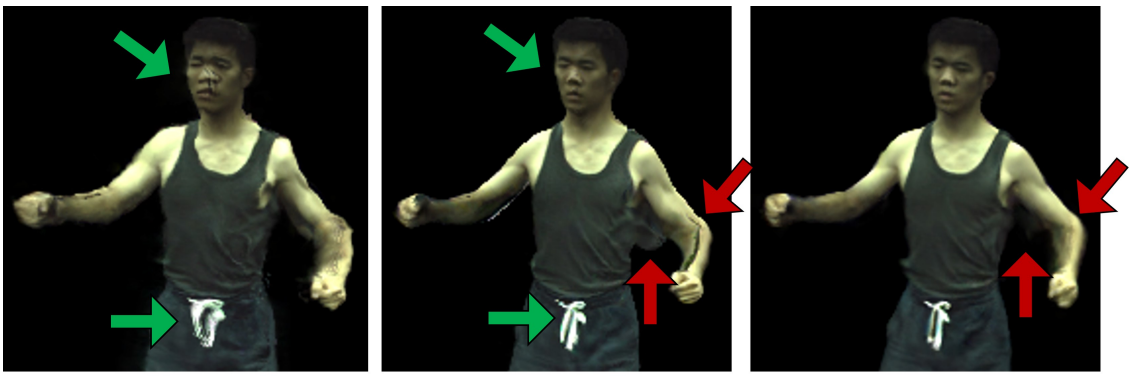
	Subject **377**			Subject **392**			Subject **393**		
	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓
HumanNeRF	29.59	0.9721	33.53	30.38	0.9626	51.03	27.56	0.9535	55.69
Ours (Separate)	29.61	0.9734	27.66	29.48	0.9640	42.65	27.28	0.9537	47.53
Ours (Single)	29.55	0.9737	26.62	30.03	0.9665	38.79	27.59	0.9558	46.16

Table C.1: Comparison on ZJU-Sparse dataset (10 images per subject).

and second-best values.

C.2.2 Results on ZJU-Sparse dataset

Table C.1 shows comparisons on the ZJU-Sparse dataset. Our method, Ours (Single), outperforms HumanNeRF and the separate-network version of our approach, Ours (Separate), in SSIM and LPIPS, with the largest margins in LPIPS, a better measure of visual quality as seen in Fig. C.3).



(a) HumanNeRF (b) Ours (Separate Nets) (c) Ours (Single Net)

Figure C.3: Our approach enhances details in the face and clothing (green). Single network training further improves shape and appearance consistency (red).

C.2.3 Results on ZJU-Dense dataset

	PSNR ↑	SSIM ↑	LPIPS* ↓
HumanNeRF	29.92	0.9684	30.97
Ours	29.81	0.9692	30.40
Ours w/o reg.	29.98	0.9700	28.47

Table C.2: Our method outperforms HumanNeRF on the ZJU-Dense dataset (an entire video per subject). The best quality is achieved when the regularization designed for sparse input is removed

We conducted an analysis on the ZJU-Dense dataset. As shown in Table C.2, our method, which was designed for sparse inputs, still demonstrates improvement. The improvement is par-

ticularly noticeable when we remove the regularization designed for handling sparse observations, indicating that the shared latent space is a promising area for exploration even for dense video.

C.2.4 Ablation Study of Photo Numbers

In addition, we analyzed how the performance is affected by the number of training images. We do see improvement with more photos on ZJU-MoCap dataset, though with diminishing returns. Table C.3 shows numerical results.

# of images per subject	PSNR ↑	SSIM ↑	LPIPS* ↓
a video ($\sim$ 600 frames)	29.81	0.9692	30.40
20 images	29.45	0.9679	32.38
10 images	29.06	0.9653	37.19

Table C.3: The ablation study of photo numbers run on ZJU-MoCap.

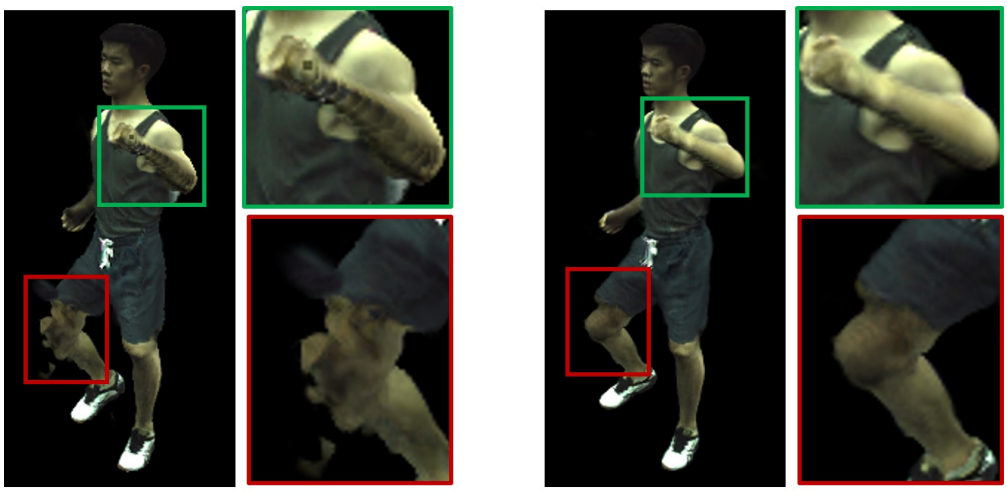
C.2.5 Novel Pose Evaluation

	PSNR ↑	SSIM ↑	LPIPS* ↓
Separate Nets	29.08	0.9691	31.05
Single Net	30.06	0.9727	27.75

Table C.4: We achieve better performance for unseen poses when training all photos with different appearances in a single network.

Our focus was on maintaining original poses, not re-posable avatar creation, avoiding, e.g., making a famous tennis player perform actions they never did. That said, experiments suggest that our method is capable of handling poses that have not been previously encountered, especially

when all photos are trained within a single network. We performed an analysis on ZJU-Sparse dataset (10 frames per subject) where we applied the learned model to body poses from the unseen frames (∼600 frames per subject). As presented in Table C.4, single-network training achieves better performance in all metrics. This is because the optimized single, universal motion weight volume can be constrained by a much larger number of poses compared to the separate ones, resulting in a better solution. Fig. C.4 shows the visual comparison.



(a) Separate Nets (b) Single Net

Figure C.4: Single-network training improves appearance consistency (green) and maintains body shapes (red) for unseen poses.