UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Thiago Luange Gomes

# TRANSFERÊNCIA DE MOVIMENTO E APARÊNCIA HUMANA ENTRE VIDEOS MONOCULARES

Belo Horizonte

2021

Thiago Luange Gomes

# TRANSFERÊNCIA DE MOVIMENTO E APARÊNCIA HUMANA ENTRE VIDEOS MONOCULARES

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Erickson Rangel do Nascimento
Coorientador: Renato José Martins

Belo Horizonte
2021

Thiago Luange Gomes

# TRANSFERRING HUMAN MOTION AND APPEARANCE IN MONOCULAR VIDEOS

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Erickson Rangel do Nascimento
Co-Advisor: Renato José Martins

Belo Horizonte
2021

Gomes, Thiago Luange

G633t Transferring human motion and appearance in monocular vídeos [manuscrito] / Thiago Luange Gomes. — Belo Horizonte, 2021.
103 f. : il. ; 29cm

Orientador: Erickson Rangel do Nascimento.
Coorientador: Renato José Martins.

Tese (doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.
Referências: f. 94-103

1. Computação – Teses. 2. Visão por computador - Teses. 3. Computação gráfica - Teses. 4. Processamento de videos – Teses. 5. Corpo humano e tecnologia em movimento – Teses. I. Nascimento, Erickson Rangel do. II. Martins, Renato José. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. IV. Título.

CDU 519.6*82.10(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz - CRB 6ª Região nº 819.

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

# FOLHA DE APROVAÇÃO

Transferring Human Motion and Appearance in Monocular Videos

**THIAGO LUANGE GOMES**

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. Erickson Rangel do Nascimento - Orientador
Departamento de Ciência da Computação - UFMG

Prof. Renato José Martins - Coorientador
Escola de Engenharia - Universidade da Borgonha

Prof. Mario Fernando Montenegro Campos
Departamento de Ciência da Computação - UFMG

Prof. Manuel Menezes de Oliveira Neto
Instituto de Informática - UFRGS

Prof. Anderson de Rezende Rocha
Instituto de Computação - UNICAMP

William Robson Schwartz
Prof. William Robson Schwartz
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 6 de Agosto de 2021.

*Eu dedico esse trabalho primeiramente a minha família, em especial ao meu filho Gabriel e a minha esposa Nayara Vilela Avelar, que me deu apoio e incentivo durante todo o processo, dedico também a todos as pessoas que participaram diretamente e indiretamente do processo, Prof. Erickson Rangel do Nascimento, Prof. Renato José Martins, João Pedro Moreira Ferreira, Rafael Azevedo, Guilherme Torres, Thiago Coutinho e muitos outros amigos.*

# Agradecimentos

Quando eu olho para trás, eu lembro de um monte de pessoas que influenciaram meu trabalho e minha vida. Primeiro de tudo, quero agradecer ao meu orientador Prof. Erickson Rangel do Nascimento, sendo um prazer trabalhar em seu grupo e seu estilo de pesquisa vai me influenciar em toda minha carreira. Ele me apresentou a área de visão computacional, pela qual eu criei paixão por trabalhar. Segundo, eu quero agradecer ao meu coorientador Prof. Renato José Martins que ajudou muito com seu conhecimento, trabalho e incentivo. Meus agradecimentos especiais são para João Pedro Moreira Ferreira e os integrantes do Verlab por ouvir ideias, problemas da vida real e científica e ajudar na construção deste trabalho. Eu também quero agradecer aos outros professores que ministraram matérias importantes para minha formação no doutorado e ao departamento por fornecer apoio e condições extraordinárias de trabalho.

Deixo um agradecimento necessário à Universidade Federal de Ouro Preto que me contemplou com o afastamento e tornou possível o desenvolvimento do projeto. Agradeço também à CAPES, CNPq e FAPEMIG por financiar o projeto de diferentes formas, como participação em conferência e compra de equipamentos.

Um agradecimento muito especial a todos meus amigos. Vocês sempre acreditaram em mim e me ajudaram a não perder as perspectivas. No entanto, eu sou de longe mais grato à minha família e em especial à minha esposa, Nayara Vilela Avelar que me deu mais amor e incentivo do que eu posso pagar de volta!

*"Perception is strong and sight weak. In strategy it is important to see distant things as if they were close and to take a distanced view of close things."*
(Miyamoto Musashi)

# Resumo

Esta tese está no contexto de transferência de movimento e aparência humana entre vídeos monoculares com preservação de características do movimento, forma do corpo e qualidade visual. Em outras palavras, dados dois vídeos de entrada, esta tese investiga como sintetizar um novo vídeo, onde a pessoa do primeiro vídeo é colocada no contexto do segundo vídeo realizando os movimentos da pessoa do segundo vídeo. Possíveis domínios de aplicação são filmes e anúncios que contam com personagens sintéticos e ambientes virtuais para criar conteúdo visual. Este trabalho introduz dois novos métodos para transferir aparência e movimento humano entre vídeos monoculares e por consequência aumentar as possibilidades criativas de conteúdo visual. Ao contrário dos recentes métodos de transferência baseados em aprendizado, nossas abordagens levam em conta restrições de forma, aparência e movimento tridimensional. Especificamente, o primeiro método usa uma nova técnica de renderização baseada em imagens que apresenta resultados comparáveis com as técnicas mais modernas, com a vantagem de não demandar um custoso processo de treinamento. O segundo método faz uso de técnicas de renderização diferencial e modelos paramétricos para produzir um modelo 3D completamente controlável, ou seja, um modelo onde o usuário pode controlar a pose humana e os parâmetros de renderização. Experimentos em diferentes vídeos mostram que nossos métodos preservam características específicas do movimento que devem ser mantidas (por exemplo, pés tocando o chão e mãos tocando um objeto) enquanto mantém os melhores valores para aparência em termos de Similaridade Estrutural (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), Erro Quadrático Médio (EQM) e Fréchet Video Distance (FVD). Além disso, como resultado adicional, esta tese apresenta uma base de dados composta de vídeos com anotações das restrições do movimento e movimento pareados para avaliar a transferência de movimento.

**Palavras-chave:** Transferência de Movimento Humano, Síntese de aparência humana, Síntese de Vídeo, Síntese Visual e Manipulação de Imagens.

# Abstract

This dissertation is in the context of transferring human motion and appearance from video to video preserving motion features, body shape, and visual quality. In other words, given two input videos, we investigate how to synthesize a new video, where a target person from the first video is placed into a new context performing different motions from the second video. Possible application domain are movies and advertisements that rely on synthetic characters and virtual environments to create visual content. We introduce two novel methods for transferring appearance and retargeting human motion from monocular videos, and by consequence, increase the creative possibilities of visual content. Differently from recent appearance transferring methods, our approaches take into account 3D shape, appearance, and motion constraints. Specifically, our first method is based on a hybrid image-based rendering technique that exhibits competitive visual retargeting quality compared to state-of-the-art neural rendering approaches, even without computationally intensive training. Taking advantages of both differentiable rendering and the 3D parametric model, our second data-driven method produces a fully 3D controllable human model, *i.e.*, the user can control the human pose and rendering parameters. Experiments on different videos show that our methods preserve specific features of the motion that must be maintained (*e.g.*, feet touching the floor, hands touching a particular object) while holding the best values for appearance in terms of Structural Similarity (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), Mean Squared Error (MSE), and Fréchet Video Distance (FVD). We also provide to the community a new dataset composed of several annotated videos with motion constraints for retargeting applications and paired motion sequences from different characters to evaluate transferring approaches.

**Palavras-chave:** Motion Transfer, Human Motion, Motion Retargeting, Human-image synthesis, Video Generation, Image Synthesis and Image Manipulation.

# List of Figures

1.1 Example of the manual process of creating synthetic characters and virtual environments. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

1.2 Common tasks to create visual content. . . . . . . . . . . . . . . . . . . . . 19

1.3 Overview of the motion and appearance transfer from a target video to different videos. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

1.4 Overview of our data-driven formulation for transfer appearance and reenact human actors. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

2.1 Examples of the provided annotations to 2D human pose estimation. . . . 28

2.2 Different pose representations for the human body. . . . . . . . . . . . . 29

2.3 Example results on skeleton and 3D human body shape estimation of Sigal et al. [2007] (Source: Sigal et al. [2007].) . . . . . . . . . . . . . . . . . . 31

2.4 Example results on skeleton and 3D human body shape estimation of Bogo et al. [2016]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

2.5 Example results on shape and pose estimation of Kanazawa et al. [2018]. . 33

2.6 Transferring a walk motion to characters with different skeletons (Source: Gleicher [1998]). . . . . . . . . . . . . . . . . . . . . . . . . . 34

2.7 Esser et al. [2018]'s model to synthesize new images based on estimated edges and body joint locations. . . . . . . . . . . . . . . . . . . . . . . . 36

2.8 Wang et al. [2018]'s model to synthesize new videos based on estimated pose. 36

2.9 Motion and appearance transfer in different morphologies . . . . . . . . . . 38

3.1 Overview of our general methodology. . . . . . . . . . . . . . . . . . . . . 42

3.2 The initial mesh and pose in our motion representation using the SMPL model. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

3.3 Example of global pose reconstruction. The input images cropping (blue), the resulting change of field of view (red), and the global camera coordinates (black). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3.4 Example of 3D constraints from human-to-object interactions. . . . . . . . 48
3.5 Example of hybrid 2D/3D constraints from human-to-object interactions. . 48
3.6 Example of the intermediate results of the compositing step. . . . . . . . . 49
3.7 Overview of our Image-Based Rendering approach. . . . . . . . . . . . . . 51
3.8 Semantic guided deformation. . . . . . . . . . . . . . . . . . . . . . . . . 53
3.9 Our common UV texture map space. . . . . . . . . . . . . . . . . . . . . 54
3.10 Rendering of the visibility maps and texture images. . . . . . . . . . . . . 55
3.11 3D differentiable human rendering strategy. . . . . . . . . . . . . . . . . . 57
3.12 Images examples that compose our loss functions in method II. . . . . . . 59
3.13 Our U-net architecture. The U-Net is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks. . . . . 60
3.14 Schematic representation of our discriminators' architecture: the Image Discriminator (**a**) and Face Discriminator (**b**). . . . . . . . . . . . . . . . . . 61

4.1 Human retargeting dataset. . . . . . . . . . . . . . . . . . . . . . . . . . 65
4.2 The subjects participating in our dataset. . . . . . . . . . . . . . . . . . . 66
4.3 Overview of all motions and subjects presented in our proposed dataset. . 67

5.1 Qualitative ablation analysis of the retargeting. . . . . . . . . . . . . . . . 75
5.2 Retargeted trajectory with motion constraints. . . . . . . . . . . . . . . . . 75
5.3 Method I: Qualitative analysis in the dataset sequences. . . . . . . . . . . 78
5.4 Method I: Qualitative evaluation to bruno-mars sequence. . . . . . . . . . 80
5.5 Method I: Qualitative evaluation to tom-cruise sequence. . . . . . . . . . . 81
5.6 Method I: Qualitative evaluation on a sequence from the iPER dataset. . . 82
5.7 Method I: Model-to-virtual. . . . . . . . . . . . . . . . . . . . . . . . . . 83
5.8 Method II: Ablation study. . . . . . . . . . . . . . . . . . . . . . . . . . . 85
5.9 Method II: Qualitative comparison. . . . . . . . . . . . . . . . . . . . . . . 88
5.10 Method II: Human retargeting example. . . . . . . . . . . . . . . . . . . . 89

6.1 Limitations. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92

# List of Tables

5.1 **Ablation study**. Comparison of mean MSE and FVD for different ablated versions of our method on all motion types of our dataset (best in bold). . 74

5.2 **Quantitative ablation analysis of the retargeting with different pair of actors.** Table shows the error in pixels between the constraints position and the end-effectors location. This error indicates how far the end-effector is from the target position (better close to 0). . . . . . . . . . 74

5.3 **Quantitative ablation analysis of the retargeting with different motion types.** Table shows the error in pixels between the constraints position and the end-effectors location. This error indicates how far the end-effector is from the target position (better close to 0). . . . . . . . . . 74

5.4 **Visibility maps and semantic-guidance analysis**. Average forgery performance for each movement (better close to 0). . . . . . . . . . . . . . . . 76

5.5 **Method I: Comparison with state of the art**. SSIM, LPIPS, MSE, and FVD comparison by motion types and pair of actors from our dataset (best in bold, second-best in italic). . . . . . . . . . . . . . . . . . . . . . . 77

5.6 **Forgery performance**. Quantitative metrics of the movement transfer approaches considering the forgery performance metric (better close to 0). 79

5.7 **Method I: Comparison with iPER in their proposed dataset**. Our approach is able to provide the best values in terms of SSIM and LPIPS, while performing worse in terms of FVD (best in bold). . . . . . . . . . . 83

5.8 **Method II: Ablation study**. SSIM, LPIPS, MSE, and FVD comparison by motion types. Best in bold. . . . . . . . . . . . . . . . . . . . . . . . . . 85

5.9 **Method II: Comparison with state of the art**. SSIM, LPIPS, MSE, and FVD comparison by motion types and pair of actors from our dataset (best in bold, second-best in italic). . . . . . . . . . . . . . . . . . . . . . . 87

# List of Acronyms

**CNN** Convolutional Neural Network. 14

**DR** Differentiable renderers. 22

**FK** Forward kinematics. 27

**GAN** Generative Adversarial Network. 18

**GCN** Graph Convolutional Networks. 21, 23

**IK** Inverse kinematics. 17

**SCAPE** Shape Completion and Animation for PEople. 13

**SMPL** Skinned Multi-Person Linear Model. 12, 14

# List of Symbols

$\mathbf{K}$ Camera intrinsic matrix. 29

$\mathcal{M}$ The orthonormal principal components of shape displacements and the blend skinning weights. 27

$\mathbf{P}^i$ The pose of a joint $i$. 27

$\Theta$ the subset of inlier joints. 29

$\boldsymbol{\beta}^{\boldsymbol{s}}$ Vector of shape parameters of the source model. 28

$\boldsymbol{\beta}$ Vector of shape parameters. 26

$\boldsymbol{\theta}$ Vector of pose parameters. 27

$\mathbf{t}$ Camera translation. 29

# Contents

**1 Introduction** **18**

1.1 Contextualization . . . . . . . . . . . . . . . . . . . . . . . . . 20

1.2 Problem Definition . . . . . . . . . . . . . . . . . . . . . . . . 21

1.3 Dissertation Statement . . . . . . . . . . . . . . . . . . . . . . 22

1.4 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

1.5 Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

1.6 Disclaimer . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

**2 Related Work** **27**

2.1 3D Human pose and shape estimation . . . . . . . . . . . . . . . 27

2.2 Mesh Reconstruction . . . . . . . . . . . . . . . . . . . . . . . 32

2.3 3D Retargeting Motion . . . . . . . . . . . . . . . . . . . . . . 33

2.4 Synthesizing Views . . . . . . . . . . . . . . . . . . . . . . . . 35

2.5 Graph Convolutional Networks (GCN) and Adversarial Learning . . . . 39

2.6 Differentiable Rendering . . . . . . . . . . . . . . . . . . . . . 39

2.7 Summary and Closing Remarks . . . . . . . . . . . . . . . . . . 40

**3 Methodology** **41**

3.1 General Methodology . . . . . . . . . . . . . . . . . . . . . . . 41

3.2 Shared Components . . . . . . . . . . . . . . . . . . . . . . . . 42

3.2.1 Human Motion Estimation . . . . . . . . . . . . . . . . . . 43

3.2.2 Motion Retargeting . . . . . . . . . . . . . . . . . . . . . . 46

3.2.3 Compositing . . . . . . . . . . . . . . . . . . . . . . . . . . 49

3.3 Method I: Image-Based Rendering . . . . . . . . . . . . . . . . . 50

3.3.1 Target Character Processing . . . . . . . . . . . . . . . . . 51

3.3.2 Model Rendering and Image Compositing . . . . . . . . . . . 54

3.4 Method II: 3D Differentiable Human Rendering . . . . . . . . . . . 55

3.4.1 Target Character Processing . . . . . . . . . . . . . . . . . 56

3.5 Summary and Closing Remarks . . . . . 62

**4 Human Retargeting Dataset and Evaluation 63**

4.1 Existing Datasets . . . . . 63

4.2 Human Retargeting Dataset . . . . . 64

4.3 Protocol . . . . . 68

4.4 Evaluation Metrics . . . . . 69

4.5 Videos for Qualitative Analysis . . . . . 70

**5 Experiments and Results 72**

5.1 Comparison with Previous Approaches . . . . . 72

5.2 Method I: Image-Based Rendering . . . . . 73

5.2.1 Implementation details . . . . . 73

5.2.2 Processing Time . . . . . 73

5.2.3 Ablation Study . . . . . 73

5.2.4 Quantitative Analysis . . . . . 76

5.2.5 Qualitative Analysis . . . . . 79

5.2.6 Results in the iPER Dataset . . . . . 81

5.2.7 Discussion . . . . . 82

5.3 Method II: 3D Differentiable Human Rendering . . . . . 83

5.3.1 Implementation details . . . . . 83

5.3.2 Processing Time . . . . . 84

5.3.3 Training Setup . . . . . 84

5.3.4 Ablation Study . . . . . 85

5.3.5 Quantitative Comparison with State of The Art . . . . . 86

5.3.6 Qualitative Visual Analysis . . . . . 86

5.3.7 Discussion . . . . . 88

5.4 Conclusions and Closing Remarks . . . . . 89

**6 Conclusions 91**

6.1 Future Work . . . . . 92

**Bibliography 94**

# Chapter 1

# Introduction

Synthetic characters and virtual environments are vital components to create visual content. Nevertheless, creating these components requires a large amount of manual work wherein artists apply low-level instructions such as drawing the skeletons, manipulating polygons, edges, and vertices (see Figure 1.1). Furthermore, humans learned early in their lives to recognize human forms and make sense of what emotions and meaning are being communicated by human movement. We are, by nature, specialists in the human movement analysis. Even for a meticulous artist, it may be hard to capture the fine details of human form and motion in a purely manual approach. Small imperfections when animating virtual actors might create a false appearance, especially the high-frequency motion components in a moving character. Thus, the movement also plays a central role in synthesizing realistic virtual moving actors.

In the last years, we witnessed an overwhelming growth in the ways and quantity of visual content that we consume. We consume visual content when we watch movies, play digital games, browse the Internet, and immerse in virtual reality or augmented reality using new devices. For example, in September 2019, videos accounted for 60% of the total volume of downstream traffic on the Internet and gaming 8.0% [Sandvine, 2019]. Recent studies that take into account the effects of the global COVID-19 pandemic predict that video viewing will account for 82% of all internet traffic by 2022 [InterDigital, 2020]. The democratization of disseminating visual content mainly justifies this growth. For example, platforms such as YouTube and Steam provide tools and services helping visual content developers expose their works to the world for a decreasing price or even free.

People need to create visual content as part of this process, but unfortunately, only a few are talented enough to express themselves visually. Even for one qualified person, acquiring technical knowledge in image editing programs and modeling tools

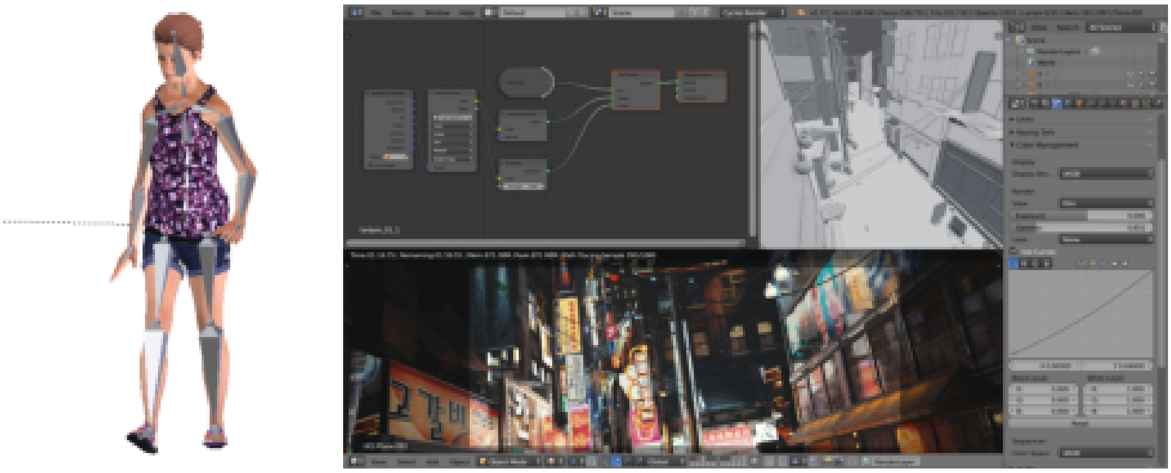

Figure 1.1: Example of the manual process of creating synthetic characters and virtual environments by manipulating polygons, edges, and vertices in simulated 3D space (Source: blender).

demands a great effort and time. Thus, there is considerable interest in tasks that assist in creating visual content based on high-level instructions, such as removing unwanted objects in personal photographs, adjusting the illumination in photographs, and inserting or changing the appearance or proportions of things in the image. Figure 1.2 shows some representative examples of the creation of visual content based on high-level instructions.

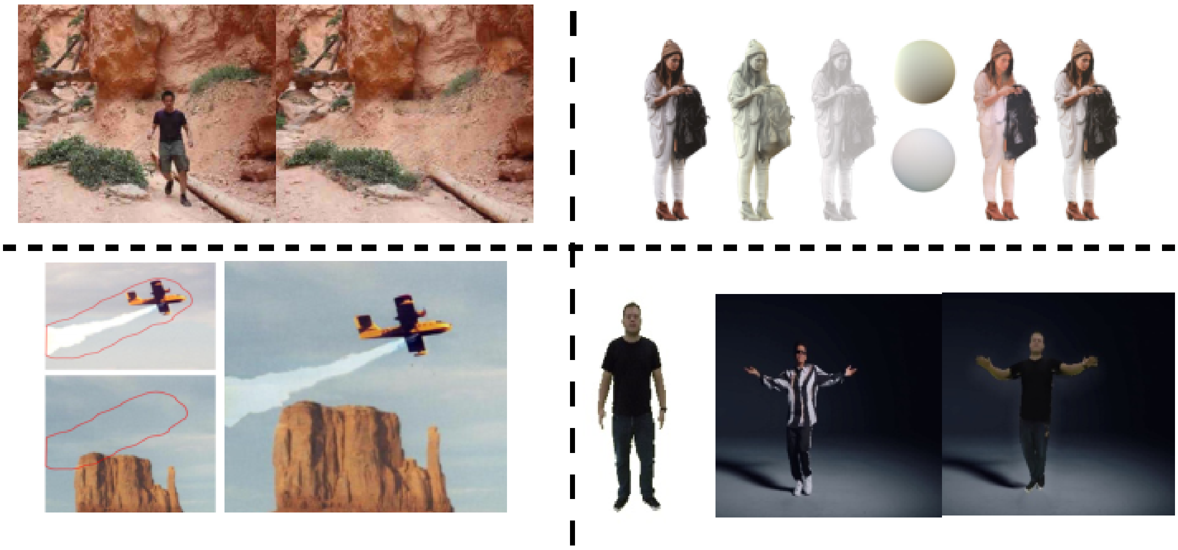

Figure 1.2: Common tasks to create visual content: *Top-left*: removing unwanted objects (image source: [Yu et al., 2018]), *Top-right*: adjusting the illumination (image source: [Kanamori and Endo, 2018]), *Bottom-left*: inserting objects (image source: [Pérez et al., 2003]) and *Bottom-right*: creating plausible videos of virtual actors from images.

In particular, the entertainment industry overcomes the difficulties of creating plausible videos of virtual actors either by a purely manual editing process or by employing specialized hardware. In both cases, the costs of production are very high. This context leads to strong business interest in researches that address the creation of human motion and synthetic characters using inexpensive hardware. Thus, an increasing number of studies apply monocular cameras as recording equipment [Chan et al., 2019; Esser et al., 2018; Wang et al., 2018]. Monocular cameras are much less expensive than commercial approaches that use 3D and 4D scanners or multi-camera studios with controlled lighting. Besides the cameras being much less costly, there are many monocular videos on the web providing a rich set of motions and appearances.

Therefore, an alternative approach to overcome the lack of high-level instruction to create visual content is to use real images (monocular videos) to drive the creation of synthetic characters and virtual environments. Nonetheless, creating plausible videos of virtual actors from images of real actors remains one of the key challenges in Computer Vision and Computer Graphics fields.

## 1.1 Contextualization

Over the past few years, the remarkable performance of the Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] has shed a new light for the problem of synthesizing faithful real-world images of humans. With the success of the GANs, we have witnessed the rise of new applications such as synthesis of human faces [Karras et al., 2018], image-to-image and video-video translation [Esser et al., 2018; Isola et al., 2017; Wang et al., 2018; Zhu et al., 2017], motion synthesis [Ferreira et al., 2021; Lee et al., 2019], and human retargeting and reenacting [Chan et al., 2019; Liu et al., 2019a; Wu et al., 2018], to name a few. Although GAN-based approaches have been achieving high quality results in generating videos and images of people, in general, they suffer with the high variability in the poses, unseen images from viewpoints not present in the training data, and they are limited to the 2D image domain.

Most recently, several methods have been proposed on body reenactment from source images [Chan et al., 2019; Esser et al., 2018; Liu et al., 2019; Ma et al., 2017; Mir et al., 2020; Sun et al., 2020]. The ultimate goal of these methods is to create a video where the body of a target person is reenacted according to the motion extracted from the monocular video. The motion is estimated considering the set of poses of a source person. Despite the impressive results for several input conditions, there are instances where most of these methods perform poorly. For instance, the works of Chan et al.

[2019] and Wang et al. [2018], only perform good reenacting of the appearance/style from one actor to another if a strict setup has complied, *e.g.*, static backgrounds, a large set of motion data of the target person to train, and actors in the same distance from the camera [Tewari et al., 2020]. Furthermore, it is hard to gauge progress from these results in the field of retargeting, as most works only report the performance of their algorithms in their own set of images which, in general, is built considering the specificities in the training regime of their approaches.

Additionally, while image-to-image translation methods are capable of generating plausible images, many applications such as Virtual Reality (VR) and Augmented Reality (AR) [Chen et al., 2017; Gallala et al., 2019; Minaee et al., 2020] require a full 3D representation of the person. The view-dependence hinders the creation of new configurations of scenes where the avatar can be included. Although view interpolation can be applied to estimate a transition between a pair of camera poses, it may create artifacts and unrealistic results when adding the virtual avatar into the new scene using unseen camera poses. Video-based rendering systems are greatly benefited through the use of realistic 3D texture-mapped models, which make possible the inclusion of virtual avatars using unrestricted camera poses and the automatic modification and re-arrangement of video footage. In addition to rendering human avatars from different viewpoints, the 3D shape also allows synthesizing of new images under different illumination conditions.

Many works in the Computer Vision and Computer Graphics communities have made great strides in capturing human geometry and motion through model-based and learning techniques. In particular, end-to-end learning approaches such as Peng et al. [2018], Kanazawa et al. [2018], and Kolotouros et al. [2019] have achieved state of the art in capturing three-dimensional motion, shape, and appearance from videos and still images from real actors. As more vision and graphics methods are integrated into new approaches such as differentiable rendering, more systems will be able to achieve high accuracy and quality in different tasks, in particular, generative methods for synthesizing videos with plausible and photo-realistic human reenactment.

## 1.2 Problem Definition

The problem addressed by this dissertation is transferring human motion and appearance from video to video preserving motion features, body shape, and visual quality. In other words, given two input videos, we investigate how to synthesize a new video, where a target person from the first video is placed into a new context performing dif-

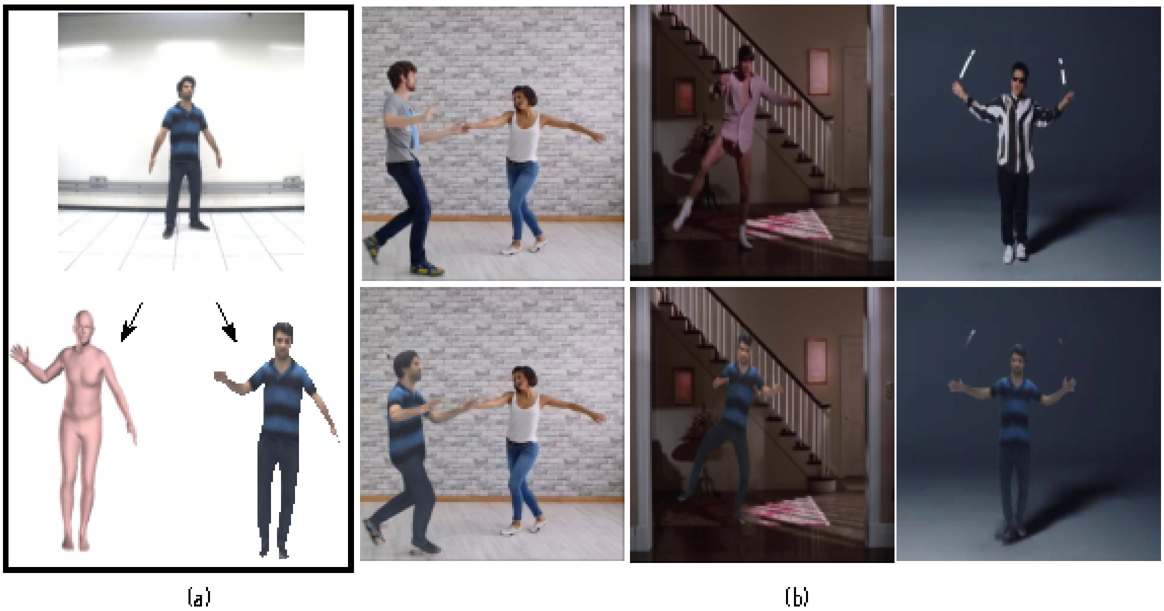

Figure 1.3: Overview of the motion and appearance transfer from a target video to different videos. After reconstructing a model for the target human (shown in (a)), we transfer his shape and motion to different videos as shown in (b). *Top row*: video with the source motion. *Bottom row*: New video with the retargeted motion and appearance of the target human model.

ferent motions from the second video (an alluring example is depicted in Figure 1.3).

## 1.3 Dissertation Statement

We state that a technique to transfer human motion and appearance from video to video can be designed considering motion constraints, body shape, and a 3D representation of people, which contributes to synthesizing more plausible videos and tackling subjects with different limb proportions and body shapes. To achieve this goal, we need to tackle the following challenges:

i. An effective approach to retargeting human motion and appearance must take into account body shape and character's interaction with the environment. Otherwise, the retarget between two people with different skeletons sizes will result in spatio-temporally inconsistency, *i.e.*, feet raised in the air, high frequency "jerkiness" in the motion, etc.;

ii. The results must retain the same quality for most poses, *e.g.*, the retargeting should not perform poorly when the person is bending;

iii. The method must provide a final representation that is compatible with traditional graphic pipelines, *i.e.*, it must generate a 3D accurate representation of people, which is a desired feature in rendering engines and games or virtual and augmented reality.

## 1.4 Contributions

In this dissertation, we present two novel video retargeting techniques for human motion and appearance transferring, which incorporate different strategies to extract 3D shape, pose, and appearance to transfer motions between two real human characters using information from monocular videos. To our best knowledge, this work is the first to transfer, not only human texture or motion but both human motion and appearance between videos, *i.e.*, we transfer motion and appearance in a unified way which allows us to tackle subjects with different limb proportions and body shape without losing the desired body proportions.

We aim to advance in the task of building a method less sensitive to the camera and poses conditions (a stable method) and overcome the lack of details. Experimental results presented later show that our approaches are both stable and shape-aware. In other words, they do not suffer from quality instability when applied in contexts slightly different from the original ones (a small difference in camera position, uncommon motions, pose translation, etc.) and they can handle different morphologies in the retargeting. Moreover, we performed experiments using a newly collected dataset containing several types of motions and actors with different body shapes and heights. Our results show that a technique applying 3D representation of people can still exhibit a competitive quality compared to recent deep learning techniques in generic transferring tests. Our approach achieved better results compared with end-to-end 2D learning methodologies such as the works of Wang et al. [2018] and Chan et al. [2019] in most scenarios for appearance metrics as structural similarity (SSIM), learned perceptual similarity (LPIPS), mean squared error (MSE), and Fréchet Video Distance (FVD).

The main technical contributions of this work are as follows:

i. A unified methodology carefully designed to transfer motion and appearance from video to video that preserves the main features of the human movement and retains the visual appearance of the target character, as shown in Figure 1.3;

ii. A retargeting technique considering physical constraints of the motion in 3D and the image domain; and a new image-based rendering technique that exhibits com-

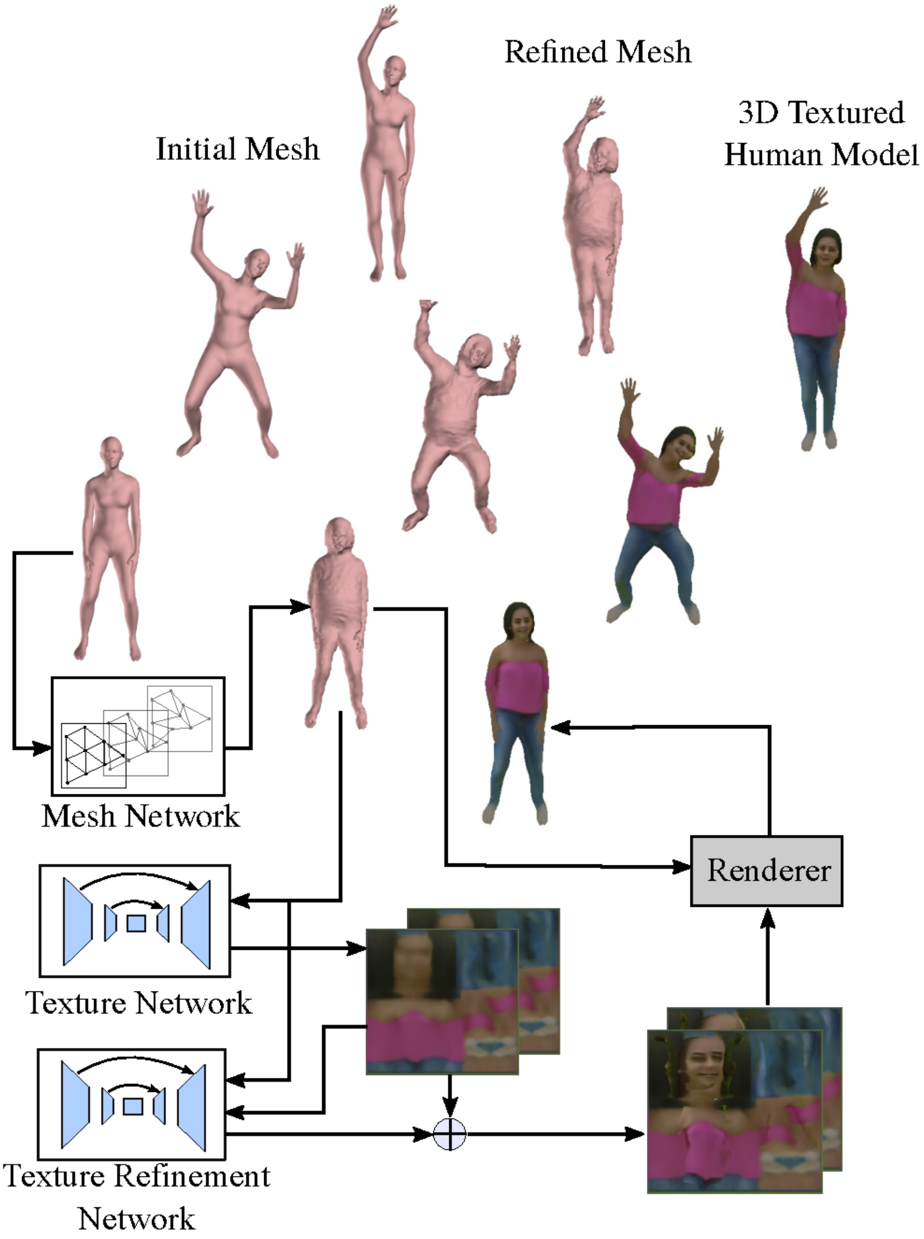

Figure 1.4: Overview of our data-driven formulation for transfer appearance and reenact human actors. Our method receives a set of frames of a person, extracts her/his mesh (left side) and outputs a fully 3D controllable human model (right side).

petitive visual retargeting quality compared to state-of-the-art neural rendering approaches, even without computationally intensive training;

iii. A novel data-driven formulation for transfer appearance and reenact human actors that produces a fully 3D controllable human model, *i.e.*, the user can control the human pose and rendering parameters, as shown in Figure 1.4;

iv. A dataset comprising several videos with annotated motion restrictions. We demonstrate the effectiveness of our approach quantitatively and qualitatively using sequences from this dataset and publicly available video sequences.

Additional contributions of our approach are: i) a graph convolutional architecture for mesh generation that leverages the human body structure information and keeps vertex consistency, which results in a refined human mesh model; and ii) a new architecture that takes advantages of both differentiable rendering and the 3D parametric model.

These contributions led to three publications:

i. Gomes, T., Martins, R., Ferreira, J., and Nascimento, E. (2020). *Do as i do: Transferring human motion and appearance between monocular videos with spatial and temporal constraints.* In WACV.

ii. Gomes, T., Martins, R., Ferreira, J., Azevedo, R., Torres, G., and Nascimento, E. (2021). *A Shape-Aware Retargeting Approach to Transfer Human Motion and Appearance in Monocular Videos.* In IJCV.

iii. Gomes, T., Coutinho, T., Azevedo, R., Martins, R., and Nascimento, E. (2022). *Creating and Reenacting Controllable 3D Humans with Differentiable Rendering.* In WACV.

The dataset and retargeting code are publicly available to the community at: https://www.verlab.dcc.ufmg.br/retargeting-motion and the presentations of our two methods are available at: https://youtu.be/seZfaiPoof4 and https://youtu.be/BkS4AsiR1es.

## 1.5 Outline

From here on, this dissertation is organized in the following chapters: Chapter 2 provides the reader an overview of the main and more recent techniques that are associated with our problem, *i.e.*, human motion estimation, motion transferring, and image synthesis. In Chapter 3, we describe the two proposed methods to transfer motion and appearance between videos. Following in Chapter 4, we present our carefully designed dataset to evaluate transferring human motion and appearance from video to video. In sequence, we present an analysis of our methodology quality in comparison to state-of-the-art techniques in Chapter 5 and discuss the limitations of our methodology. Finally, in Chapter 6, we present our conclusions and highlight future research directions.

## 1.6 Disclaimer

This document is designed to provide accurate and authoritative information in relation to the subject matter covered. The subject covered has great potential in entertainment, gaming, satire, and culture when used responsibly. Under no circumstance shall we have any liability to you for any loss or damage of any kind incurred due to the use of the information provided in this document.

# Chapter 2

# Related Work

This chapter describes the related work to our video-based retargeting problem. Despite the considerable number of works correlated to human motion analysis and human appearance synthesis, the problem addressed in this dissertation is more complex, and up to the author's knowledge, it has not been appropriately investigated in a unified transferring formulation of both human motion and appearance. Therefore, we discuss in this chapter the works that address the subproblems of our work or tackle the task of synthesizing new views of people from images, which is the closest task to our problem.

## 2.1 3D Human pose and shape estimation

In general, motion transferring approaches from video to video contains a phase of human pose/shape estimation, *i.e.*, they estimate the configuration of the human body in a given image or a sequence of images. A simple approach is to employ a 2D representation of the human body pose in the image plane [Chan et al., 2019; Esser et al., 2018; Wang et al., 2018], *i.e.*, understanding pose estimation as the problem of localizing anatomical keypoints or body joints in the image plane [Cao et al., 2017; Riza Alp Güler, 2018; Simon et al., 2017; Wei et al., 2016]. Methods of 2D human pose estimation have made significant progress during the last years, in a large extent due to the creation of large datasets [Andriluka et al., 2014; Lin et al., 2014; Riza Alp Güler, 2018] with annotated joint positions or dense correspondences from a 2D image to a 3D human shape. Figure 2.1 presents examples of 2D representations of the human body in the widely-adopted datasets.

Despite the progress in 2D human pose estimation, the 2D representation is ambiguous and brings limited information about the motion. Recent works went a step further estimating the human pose in 3D, which is a more suitable representation

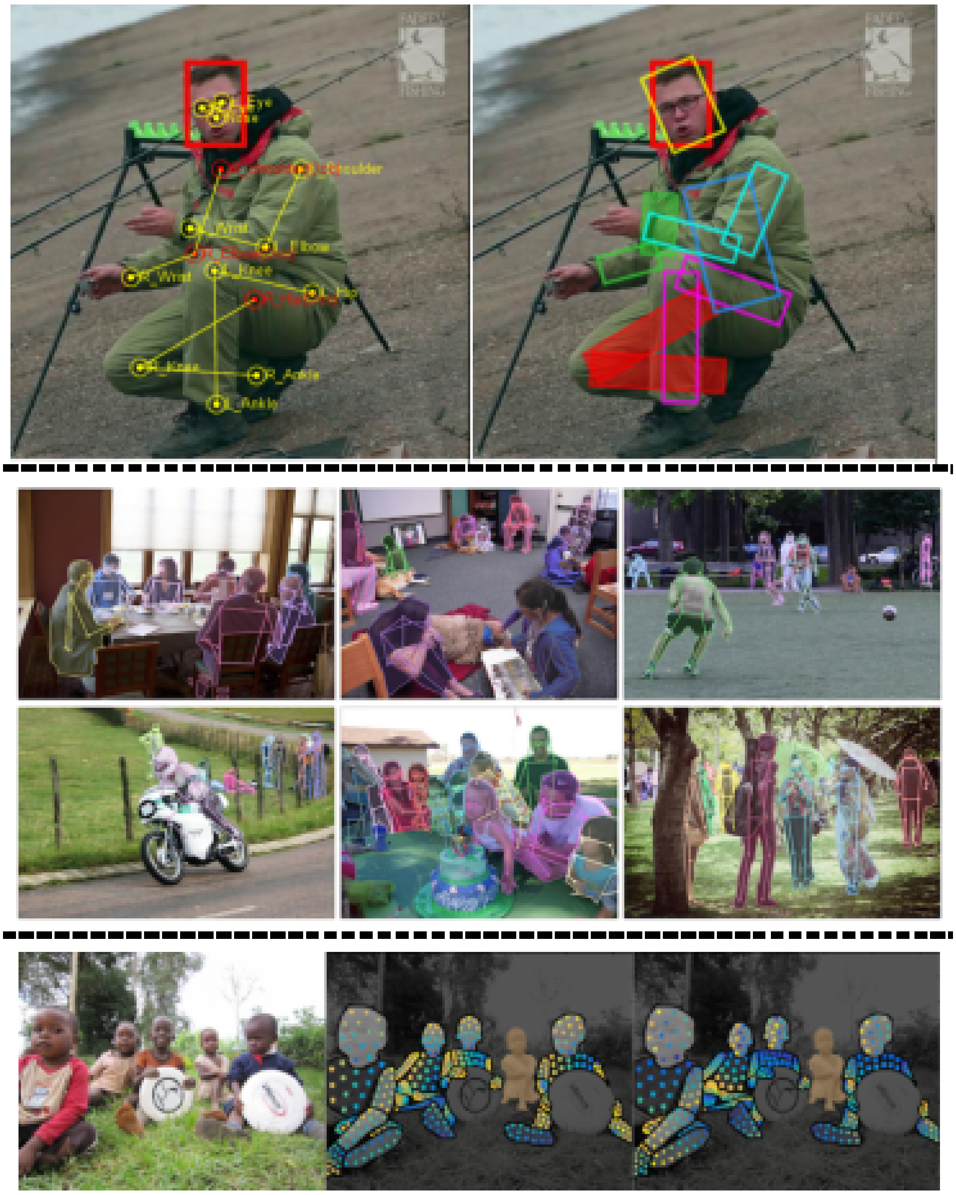

Figure 2.1: Examples of the provided annotations to 2D human pose estimation: *Top*: MPII Human Pose (Source: Andriluka et al. [2014]), *Middle*: COCO (Source: Lin et al. [2014]), *Bottom*: DensePose (Source: Riza Alp Güler [2018]).

to transfer motion since the human body movement features like velocity, acceleration, and restrictions are embedded in 3D space.

3D human pose detection from a single image is a challenging problem since the projection in the image plane produces ambiguities in images with partially occluded human poses. Moreover, an algorithm to estimate human pose must be invariant to several factors, including background scenes, lighting, clothing shape and texture, and skin color. Because of the challenges mentioned, even techniques used for 3D pose estimation have different assumptions and purposes about what should be estimated.

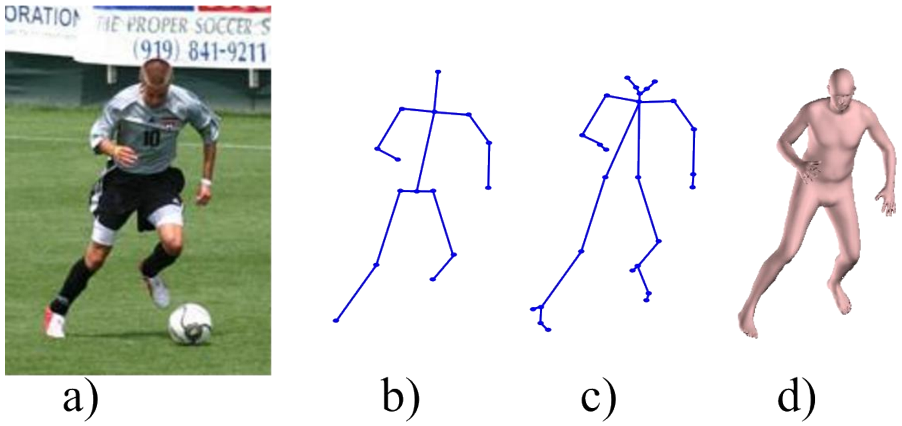

Figure 2.2: Different pose representations for the human body. a) Person; b) Kinematic joint model with 15 joint parameters; c) Kinematic joint model with 26 joint parameters; d) SMPL volumetric model (Images extracted from [Loper et al., 2015]).

One of the critical assumptions in human pose estimation is to build and describe the human body model. A good representative model should embed the human body kinematic structure information and human body shape information. In the following, we will discuss the different approaches for the problem of 3D human pose estimation using the two most used human body models: The Kinematic model representation and the Volumetric representation. Some examples of different kinematic body skeleton representations and a volumetric representation with Skinned Multi-Person Linear Model (SMPL) [Loper et al., 2015] are shown in Figure 2.2.

### 2.1.1 Kinematic Pose Model Estimation

Models that follow the skeletal structure are called kinematic chain models [Gong et al., 2016]. The set of joint positions of the kinematic model is a straightforward representation of the human body model, and it is the dominant paradigm in the field [Kanazawa et al., 2018]. Locating the major 3D joints of the body from an image is an important task in computer vision with a wide range of scientific and commercial applications, such as human-computer interaction, human-robot interaction, video surveillance, and scene understanding, to name a few.

There are different streams of work for estimating the 3D joint positions given an image. However, the main streams of work can be categorized into two approaches: two-stage and direct estimation. Two-stage methods first extract features or 2D joint locations from the image and then predict 3D joint locations by learning a function to map the features into 3D pose [Kostrikov and Gall, 2014; Tekin et al., 2015] or by regression or model fitting the 2D joints [Akhter and Black, 2015; Martinez et al., 2017].

Two-stage methods use priors to deal with pose ambiguity. Most of the priors are about the limb-length or proportions [Barron-Romero and Kakadiaris, 2001; Parameswaran and Chellappa, 2004; Ramakrishna et al., 2012].

In this context, Akhter and Black [2015] collected a dataset that includes a wide variety of stretching poses performed by trained athletes and gymnasts. Then they learned pose-dependent joint angle limits from the data and proposed a novel prior based on these joint angle limits. Two stage-methods benefit from being more robust to domain shift, but the major drawback of these methods is that their accuracy is bounded by the capacity of the 2D joints and the pose priors to explain the real poses.

There is still a lack of dataset of 3D poses for people in the wild since 3D data acquisition requires expensive setups. It is challenging to acquire data outside controlled laboratory conditions. However, after the introduction of Human3.6M dataset [Ionescu et al., 2014b], which contains 3.6 million high-resolution images with annotated 2D and 3D joint locations, many modern methods were proposed to estimate 3D joints directly from images using deep learning frameworks [Ionescu et al., 2014a; Li and Chan, 2014; Mehta et al., 2017; Zhou et al., 2016b]. Since monocular reconstruction is inherently scale-ambiguous, the input image is commonly cropped to the bounding box of the subject before 3D pose estimation [Ionescu et al., 2014a] and the output is a subject with normalized height. Li and Chan [2014] report that predicting positions relative to the parent joint of the skeleton improves the performance, but relative positions are a prohibitive simplification to motion transferring. Aware of the importance of global position and the fact that models trained only on laboratory images do not generalize well to the real world, Mehta et al. [2017] explore the use of transfer learning to leverage the highly relevant mid-and-high-level features learned on readily available in-the-wild 2D pose datasets in conjunction with the existing annotated 3D pose datasets. They reconstruct global 3D poses using a generalization of Procrustes analysis for projective alignment.

Despite the importance of 3D joint locations to many applications in computer vision, 3D joint locations are sparse and do not constrain each joint's degrees of freedom, and do not ensure that limbs are symmetric and have the correct length. Therefore, this most straightforward representation cannot be directly applied to the motion and shape transferring task.

### 2.1.2 Volumetric Pose Model Estimation

Body part volumes play an important role in the process of describing human pose [Gong et al., 2016]. A simple way of modeling a volumetric human body is to use

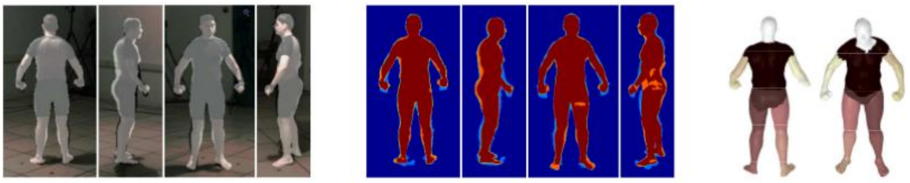

Figure 2.3: Example results on skeleton and 3D human body shape estimation of Sigal et al. [2007]. Projection of the estimated model into 4 views (left). Projection of the model onto image silhouettes (middle). Different views of the estimated 3D model (right) (Source: Sigal et al. [2007]).

geometric shapes as model components, *i.e.*, human body parts are approximated by cylinders, conics, and other shapes [Sidenbladh et al., 2000]. A more realistic way of modeling a volumetric human body is using meshes. Meshes can be deformed, which is a desirable feature for the representation of non-rigid human bodies [Sidenbladh et al., 2000].

Since the introduction of the generative model Shape Completion and Animation for PEople (SCAPE) [Anguelov et al., 2005], a data-driven method that derives the non-rigid surface deformation as a function of the pose and shape parameters, a large number of solutions were proposed in the computer vision and graphics communities to estimate both the skeleton and 3D human body shape (*e.g.*, Anguelov et al. [2005]; Hasler et al. [2010]; Loper et al. [2015]; Sigal et al. [2007]). Typically, these methods require a known segmentation and a few manual correspondences. Some illustrative examples are shown in Figure 2.3.

Bogo et al. [2016] proposed the SMPLify method, which is a fully automated approach for estimating 3D body shape and pose from 2D joints in images. SMPLify uses a Convolutional Neural Network (CNN) to estimate 2D joint locations and then fits a SMPL [Loper et al., 2015] to these joints. The fit is performed by minimizing the error between the projected joints of the model and the estimated 2D joints in the image. A few examples are shown in Figure 2.4.

In the same direction, Lassner et al. [2017b] explored the curated results from SMPLify to train 91 keypoint detectors and included an additional optimization term that accounts for the matching between the image silhouette and the 3D human silhouette contours. However, their approach requires that the segmented silhouette in the image to be consistent with the SMPL human model, *i.e.*, the silhouette in the image should be of a naked person, which is unrealistic in most videos.

In the same context, Kanazawa et al. [2018] presented a new neural-network-based approach that uses unpaired 2D keypoint annotations and 3D scans to train an end-to-

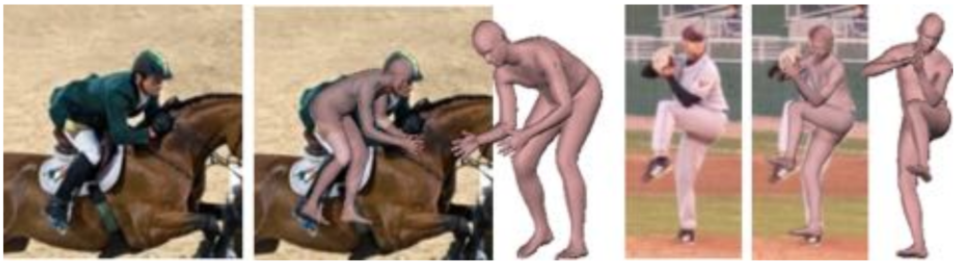

Figure 2.4: Example results on skeleton and 3D human body shape estimation of Bogo et al. [2016]. The original image (left), their fitted model (middle), and the 3D model rendered from a different viewpoint (right) (Source: Bogo et al. [2016]).

end network to infer the 3D pose/shape parameters and the camera pose. Their method outperformed the works of Bogo et al. [2016] and Lassner et al. [2017b] regarding 3D joint error and runtime. However, their approach has the drawback of always keeping the same human body shape, and simply varying the body-to-camera translation (see Figure 2.5). Kolotouros et al. [2019] combined an optimization method and a deep network to design a method less sensitive to the optimization initialization. Even though their method outperformed the works of Bogo et al., Lassner et al., and Kanazawa et al. regarding 3D joint error and runtime, their bounding box cropping strategy does not allow motion reconstruction from poses, since it frees three-dimensional pose regression from having to localize the person with scale and translation in image space. Moreover, they lack global information and temporal consistency in shape, pose, and human-to-object interactions, which are required in video retargeting with consistent motion transferring.

## 2.2 Mesh Reconstruction

Substantial advances have been made in recent years for 3D model estimation from still images. Human mesh reconstruction methods are also increasingly achieving better results as shown in works such as PiFu [Saito et al., 2019, 2020], ARCH [Huang et al., 2020], or SiCloPe [Natsume et al., 2019]. Despite the impressive results, these methods are limited to estimate static 3D character models, which require additional efforts to create animated virtual characters. In addition to the requirement that 3D models contain a skeleton hierarchy and appropriate skin weights, it is also necessary to fit a garment model into a human model in various poses. Lazova et al. [2019] automatically predict a full 3D textured avatar, including geometry and 3D segmentation layout for further generation control; however, their method cannot predict fine details and complex texture patterns.

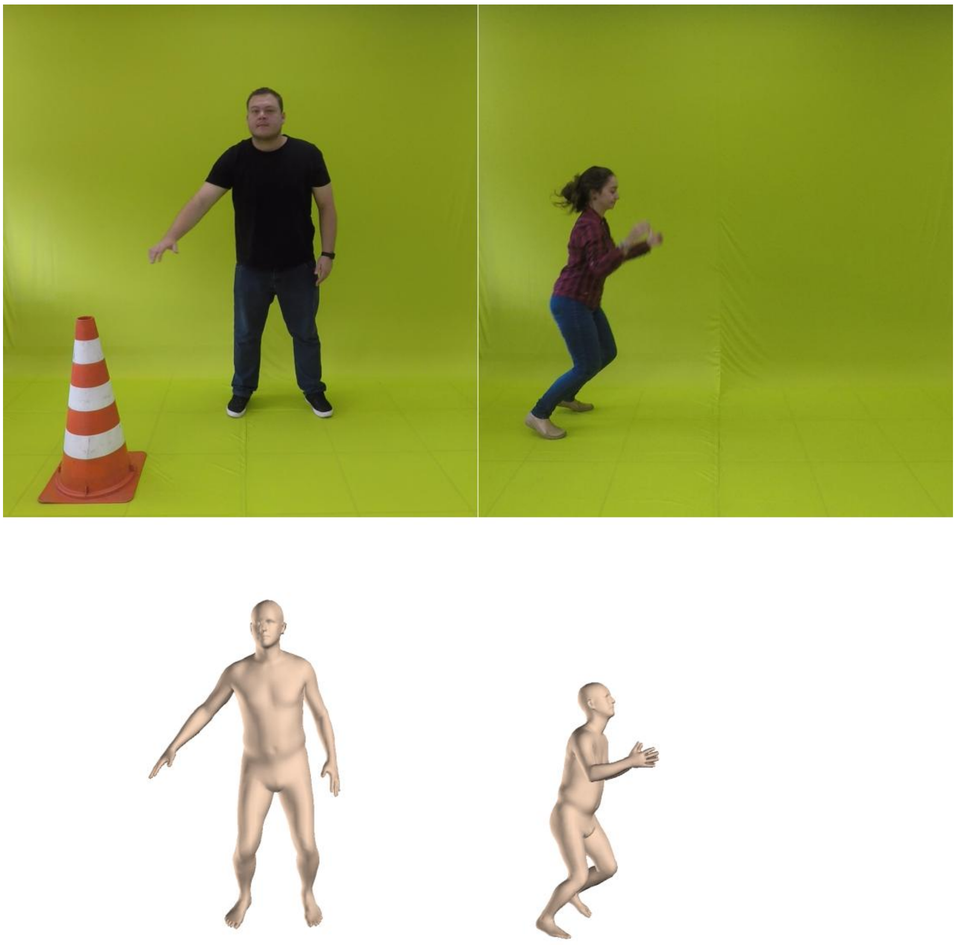

Figure 2.5: Example results on shape and pose estimation of Kanazawa et al. [2018]. *Top*: the original images. *Bottom*: ambiguous estimated models in terms of shape/translation.

## 2.3 3D Retargeting Motion

One key challenge in motion transferring approach from video to video arises from differences in the skeleton of the source and target characters. A naive transferring approach often results in spatio-temporally inconsistency, *e.g.*, feet raised in the air, high frequency "jerkiness" in the motion, etc. [Arikan and Forsyth, 2002; Gleicher, 1998, 2001; Li et al., 2002], as shown in the example of Figure 2.6.

Gleicher, in his seminal work of retargeting motion [Gleicher, 1998], faced the problem of transferring motion from one virtual actor to another. He adapted animated motions from different characters using space-time constraints, which represented the interactions of the human body segments and the environment. Lee and Shin [1999]

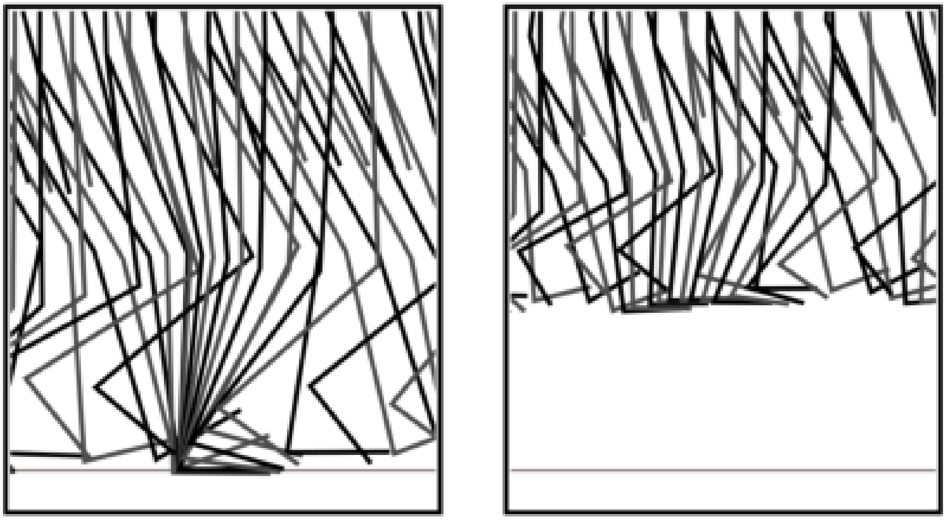

Figure 2.6: Transferring a walk motion to characters with different skeletons. *Left*: Original motion. *Right*: Applying this motion to a character that is 60% of the size of the original yields a motion that skates along horizontally above the floor (Source: Gleicher [1998]).

decomposed the problem into two stages. In the first stage, they use a kinematics solver to adjust the configuration of an articulated figure to meet the constraints in each frame. Second, to ensure smoothness, the motion displacement of every joint at each constrained frame is interpolated using multilevel B-spline curves. Tak and Ko [2005] further added dynamics constraints to perform sequential filtering to render physically plausible motions. Choi and Ko [2000] proposed an online retargeting method by solving per-frame Inverse kinematics (IK) that computes the change in joint angles corresponding to the change in end-effector positions while imposing motion similarity as a secondary task. These approaches require an iterative optimization with hand-designed activation constraints for several particular motions.

The work of Peng et al. [2018] takes a step towards in transferring motion from real people to virtual humanoids automatically. The authors proposed a reinforcement learning framework for learning full-body motion from a monocular video with real people performing skills. Despite remarkable results, their goal is to transfer the style of the motion, different from our objective that is adjust the original motion. Aberman et al. [2019] proposed a 2D motion retargeting using a high-level latent motion representation. Their method has the benefit of not explicitly reconstructing 3D poses and camera parameters, but it fails to transfer motions if the character walks towards the camera or with variations of the camera's point-of-view.

A kinematic neural network with an adversarial cycle consistency was developed in Villegas et al. [2018] to remove the manual step of defining the motion constraints. Aberman et al. [2020] explored deep learning techniques for motion retargeting between skeletons that have different structures. However, both Villegas *et al.*'s and Aberman *et al.*'s approaches are limited by disregarding scenarios where the original motion must be enhanced or adjusted because of environment changes.

## 2.4 Synthesizing Views

When working with real images of people, the retargeting problem becomes more challenging, since the deformation on the appearance starts playing a key role. Traditionally, image-based rendering techniques [Kang and Shum, 2000] have been used to solve the view synthesis problem. Image-based modeling techniques have been an alternative to traditional geometry-based methods, where a collection of sample images are used to render new views. However, several image-based rendering methods are still focused on simple objects [Kang and Shum, 2000; Shum et al., 2003; Zhang and Chen, 2004] not being well adapted for complex objects like the human body.

The past five years has witnessed the explosion of neural rendering approaches and Generative Adversarial Network (GAN). GANs have emerged as promising and effective approaches to deal with the tasks of synthesizing new views against image-based rendering approaches (*e.g.*, Kang and Shum [2000]; Shum et al. [2003]; Zhang and Chen [2004]). More recently, the synthesis of views is formulated as being a learning problem (*e.g.*, Balakrishnan et al. [2018]; Dosovitskiy et al. [2015]; Esser et al. [2018]; Tatarchenko et al. [2015]; Yang et al. [2015]), where a distribution is estimated to sample the new views. A representative approach is the work of Zhou et al. [2016a]. Their work uses a learning-based approach to implicitly approximate the geometry of the object. Instead of generating color values for each pixel in the target view, it is generated an appearance flow vector indicating the corresponding pixel in the input view that will compose the target view. However, those methods mainly synthesize rigid objects such as cars and furniture, not dealing with deformable objects with rich details such as human body.

The works proposed by Isola et al. [2017]; Mirza and Osindero [2014] showed a new approach to generate images of desired properties based on the input. In this context, Ma et al. [2017] proposed to transfer the appearance of a person to a given pose in two steps. First, their method applies a variant of the U-Net [Esser et al., 2018] focusing on the global structure of the human body. Then, they use a variant of Deep Convolutional GAN (DCGAN) to improve appearance details based on the first stage

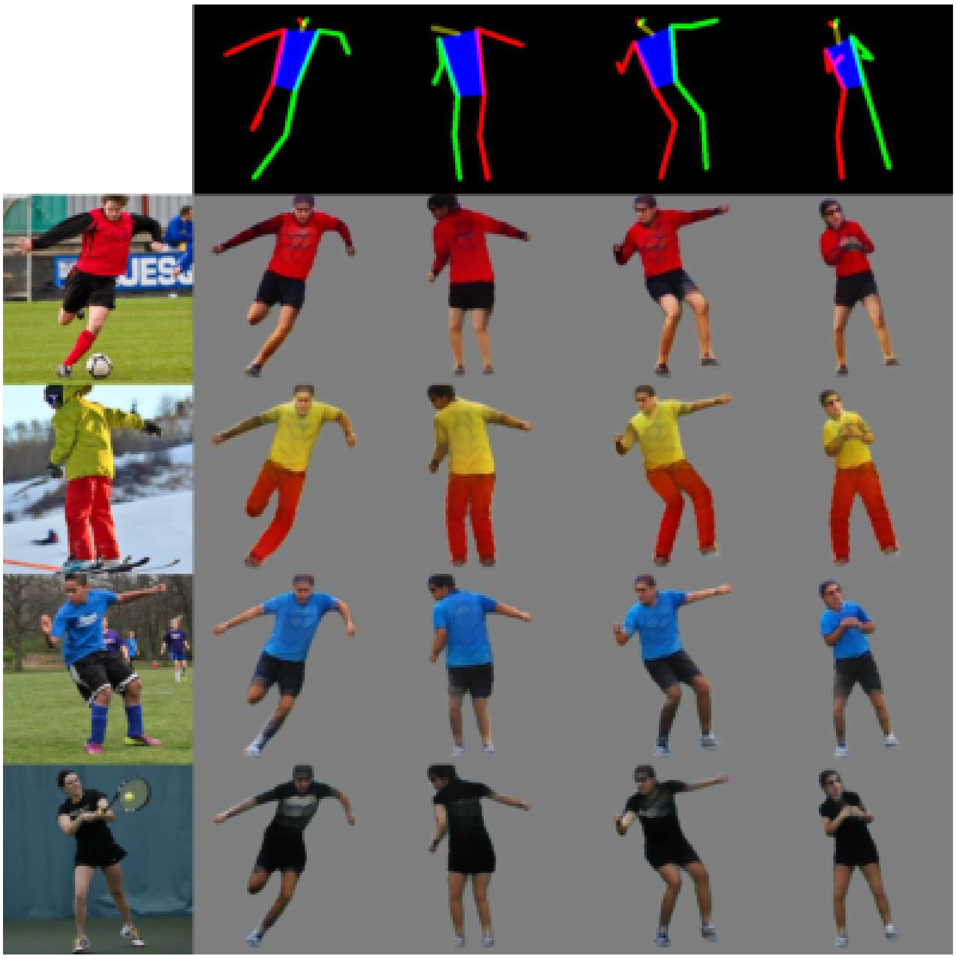

Figure 2.7: Esser et al. [2018]'s model to synthesize new images based on estimated edges and body joint locations. The Esser et al. [2018]'s model learns to infer appearance from the queries images (on the left) and can synthesize images with that appearance in different poses, as shown in the top row (Source: Esser et al. [2018]).

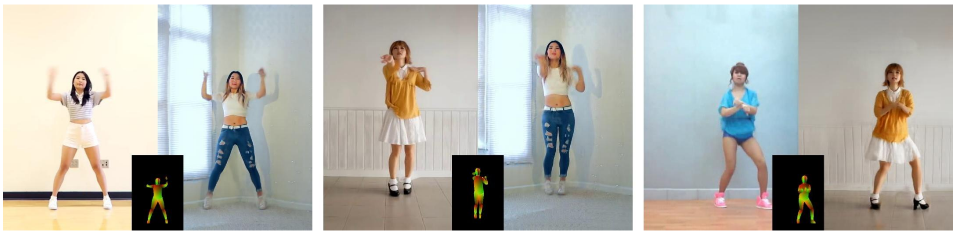

Figure 2.8: Wang et al. [2018]'s model to synthesize new videos based on estimated pose. Each set shows the original dancer, the extracted poses, and the synthesized frames (Source: Wang et al. [2018]).

result. Similarly, Lassner et al. [2017a] proposed a GAN called ClothNet. ClothNet produces random people with similar pose and shape in different clothing styles given a synthetic image silhouette of a projected 3D body model.

Balakrishnan et al. [2018], for their turn, decomposed the problem of synthesizing new views into a background and foreground layer segmentation. First, they segment the source image into a background layer and multiple foreground layers corresponding to different body parts. The segmentation allows their methodology to spatially move the body parts to target locations. The second subtask consists in composing the multiple foreground layers and background to produce the final output image. In the work of Esser et al. [2018], a conditional U-Net is used to synthesize new images based on estimated edges and body joint locations. Despite the impressive results for several inputs, learning-based methods are limited to synthesize detailed body parts such as faces, as shown in Figure 2.7.

Recent works such as Aberman et al. [2018] and Chan et al. [2019] start applying adversarial training to map 2D poses to the appearance of a target subject. Although these works employ a scale-and-translate step to handle the difference in the limb proportions between the source skeleton and the target, their synthesized views still have clear gaps in the test time compared with the training time. Wang et al. [2018] proposed a general video-to-video synthesis framework based on conditional GANs to generate high-resolution and temporally consistent videos of people, as shown in Figure 2.8.. Shysheya et al. [2019] attempt to handle the poor generalization by training a model using different actors' point-of-views. Their approach is also data-driven, which requires training a model for each new character, including the full acquisition setup information and camera poses. Mir et al. [2020] proposed to leverage the information from DensePose to learn a model to perform texture transfer of garments. Although their method is texture agnostic and not actor specific, it is designed to deal with garments transference (shirts and pants) and does not address the problem of full-body transference neither handle the cross-transference. It also disregards the motion constraints and human-to-object interactions [Hassan et al., 2019]. In the same line, Neverova et al. [2018] investigated a combination of surface-based pose estimation and deep generative models; however, their method only considers the layout locations and ignores the personalized shape and limb (joint) rotations. Despite the impressive results for several inputs, end-to-end learning-based techniques still fail to synthesize the human body's details, such as face and hands. Furthermore, it is worth noting that these techniques focus on transferring style, which leads to undesired distortions when the characters have different morphologies (proportions or body parts' lengths). An alluring example is depicted in Figure 2.9, where we perform the transfer between actors with differences in body shape (first row) and height (second row).

Another limitation of recent approaches such as Aberman et al. [2018], Chan et al. [2019], and Wang et al. [2018] is that they are data-driven, *i.e.*, they require

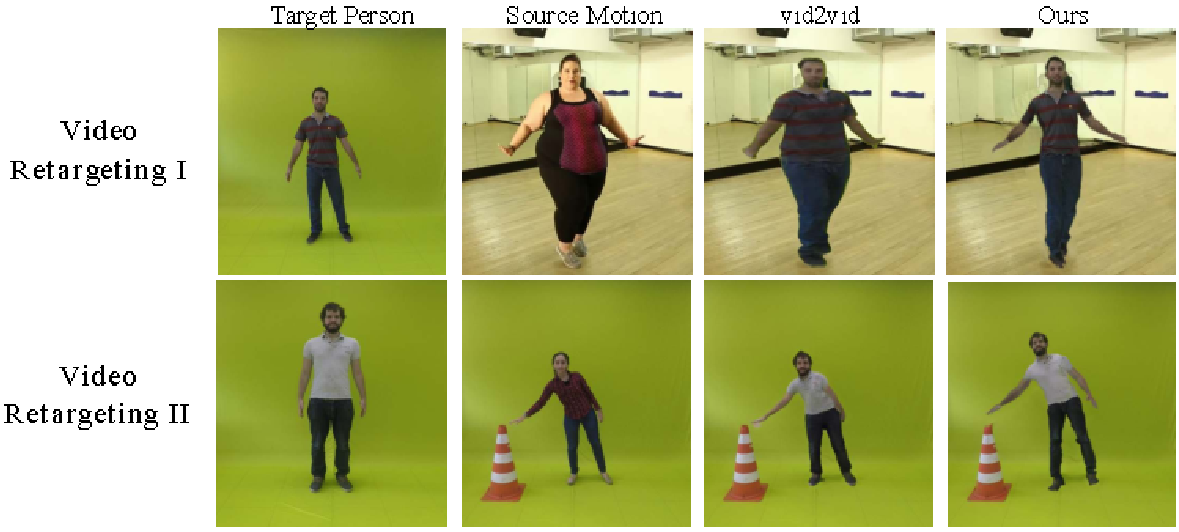

Figure 2.9: Motion and appearance transfer in different morphologies. From left to right: target person, source motion video with a human of different body shape, vid2vid [Wang et al., 2018], and our retargeting results. Note that vid2vid stretched, squeezed, and shranked the body forms whenever the transferring characters have different morphologies.

training a different GAN for each video of the target person with different motions to perform the transferring. This training is computationally intensive and takes several days on a single GPU. In order to overcome these limitations, Liu et al. [2019] proposed a 3D body mesh recovery module to disentangle the pose and shape; however, their performance significantly decreases when the source image comes from a different domain from their dataset, indicating that they are also affected by poor generalization to camera viewing changes. Recently, Sun et al. [2020] also proposed a data-driven method where projections of the reconstructed 3D human model are used to condition the GAN training, in order to maintain the structural integrity of the transfer to different poses. Nevertheless, all analyses were made in a strict setup where the person is standing parallel to the image plane, and the considered motions have reduced lateral translations.

In this dissertation, we show that there is still a performance gap of recent end-to-end deep learning techniques against an image-based model when this comprises carefully designed steps for human shape and pose estimation and retargeting. These results extend the observation of the works of Bau et al. [2019] and Wang et al. [2020b], where the authors observed that GANs still present limited generation capacity. While Bau et al. [2019] showed that generative network models could ignore classes that are too hard at the same time producing outputs of high average visual quality, Wang et al. [2020b] demonstrated that CNN-generated images are yet surprisingly easy to spot.

## 2.5 Graph Convolutional Networks (GCN) and Adversarial Learning

Graphs can be used to model many types of problems in real-world applications, including social analysis, traffic prediction, path optimization algorithms, and many more. According to Zhang et al. [2018b], by representing the data as graphs, the structural information can be encoded to model the relations among entities and furnish more promising insights underlying the data. For example, in a human skeleton, nodes are the body joints, and edges represent the limbs. In addition to the spatial information provided by the joints' position, the graph structure describes the relationship between the joints.

In this context, Graph Convolutional Networks (GCN) recently emerged as a powerful tool for learning from data lying on manifolds beyond $n$-dimensional Euclidean vector spaces. They have been widely adopted to represent 3D point clouds such as PointNet [Qi et al., 2017] or Mesh R-CNN [Gkioxari et al., 2019], and notably, to model the human body structure with state-of-the-art results in tasks such as human action recognition [Yan et al., 2018], pose estimation [Wang et al., 2020a; Zhao et al., 2019], and human motion synthesis [Ferreira et al., 2021; Ren et al., 2020; Yan et al., 2019]. Often GCNs have been combined and trained in adversarial learning schemes, as in human motion [Ferreira et al., 2021], and pose estimation [Kanazawa et al., 2018]. Our work leverages these capabilities from GCNs and adversarial training to estimate 3D texture-mapped human models.

## 2.6 Differentiable Rendering

Most methods to estimate human body geometry and texture rely on supervised training regimes and costly annotations, which makes collecting the data challenging and expensive. Thus, there are recent efforts towards leveraging 2D information and differing levels of supervision for 3D scene understanding. One of the approaches is integrating graphical rendering processes into neural network pipelines.

Differentiable renderers (DR) are operators allowing the gradients of 3D objects to be calculated and propagated through images while training neural networks. As stated in Kato et al. [2020], DR connects 2D and 3D processing methods and allows neural networks to optimize 3D entities while operating on 2D projections. Loper and Black [2014] introduced an approximate differentiable render that generates derivatives from projected pixels to the 3D parameters. Kato et al. [2018] approximated the backward gradient of rasterization with a hand-crafted function. Liu et al. [2019b] proposed a formulation of the rendering process as an aggregation function fusing the probabilistic contributions of all mesh triangles with respect to the rendered pixels. Niemeyer et al. [2020] represented surfaces as 3D occupancy fields and used a numerical method to find the surface intersection for each ray, then they calculate the gradients using implicit differentiation. Mildenhall et al. [2020] encoded a 3D point and associated view direction on a ray using periodic activation functions, then they applied classic volume rendering techniques to project the output colors and densities into an image, which is naturally differentiable. In this dissertation, we propose a carefully designed architecture for human neural rendering, leveraging the new possibilities offered by differentiable rendering techniques [Liu et al., 2019b; Loper and Black, 2014; Ravi et al., 2020; Zhang et al., 2020].

## 2.7 Summary and Closing Remarks

In this chapter, we presented an overview of human pose estimation methods from images and representations of the human body. The representation of the human body is one of the critical assumptions in our problem. A good representative model should embed the human body kinematic structure information and human body shape information. Since meshes are a realistic way of modeling a human body, we presented a brief overview of human mesh reconstruction methods. These methods are limited to estimating static 3D character models, which is insufficient to create animated virtual characters. In sequence, we introduced the retargeting problem and presented some relevant solutions. Following, we presented view synthesis methods that focus on transferring style without considering differences in the human body shape of the source and target characters, which leads to undesired distortions when the characters have different morphologies (proportions or body parts' lengths). Finally, we introduced GCN and Differentiable Rendering techniques. GCN is a powerful representation for learning from data lying on manifolds beyond $n$-dimensional Euclidean vector spaces, and Differentiable Rendering is a promising technique to supervision 3D scene understanding using 2D information.

# Chapter 3

# Methodology

This chapter presents two human transferring methods considering the importance of human motion, body shape, and appearance in the retargeting. Unlike most techniques that transfer either appearance [Aberman et al., 2018; Chan et al., 2019; Esser et al., 2018; Wang et al., 2018] or motion independently [Peng et al., 2018; Villegas et al., 2018], we present techniques that simultaneously considers body shape, motion retargeting constraints, and human-to-object interactions over time, while retaining visual appearance quality.

## 3.1 General Methodology

This section details the steps used to design our two new methods to transfer human motion and appearance from video to video. As depicted in the Figure 3.1, our two methodologies build upon our general methodology composed of four main components:

i. *Human Motion Estimation:* This component estimates the motion of the character performing actions in the source video, where essential aspects of plausible movements, such as a shared coordinate system for all image frames and temporal motion smoothness are ensured;

ii. *Target Character Processing:* This component extracts the target character's appearance and body shape in the second video;

iii. *Motion Retargeting:* This component adapts the estimated movement to the body shape of the target character while considering temporal motion consistency and the physical human interactions (constraints) with the environment;

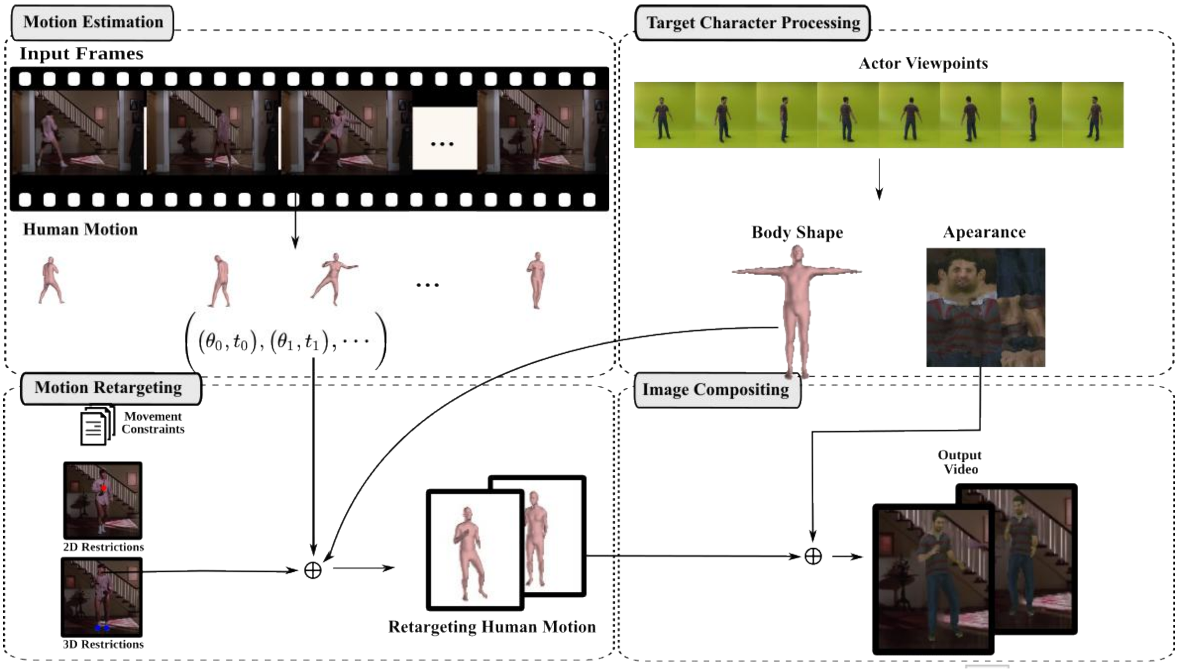

Figure 3.1: Overview of our general methodology. Each component is designed to deal with a subproblem of the video-to-video retargeting problem. The four subproblems are: human motion estimation in the source video (Human Motion Estimation); appearance and shape estimation in the target video (Target Character Processing); motion transfer from source character to target character (Motion Retargeting); and target person synthesis into the source video (Compositing).

iv. *Compositing:* This component combines the extracted target character appearance and the adapted movement into the background of the source video.

A central objective of our general methodology is to split the video-to-video retargeting problem into subproblems. Dealing with the subproblems will ensure that our retargeting methods: i) retain the same quality for most poses (Human Motion Estimation); ii) preserve visual quality (Target Character Processing); iii) take into account body shape and the character's interaction with the environment (Motion Retargeting), which allows handling different morphologies in the retargeting; and iv) place the target person into a new context (Compositing).

## 3.2 Shared Components

In this section, we detail three components shared by the novel video retargeting techniques.

### 3.2.1 Human Motion Estimation

As discussed in the previous chapter, for most motion transferring approaches from video to video, it is necessary to estimate the configuration of the human body in a given image or a sequence of images. Unlike recent learning-based techniques, we do not employ a 2D representation to this task, since 2D representations have two drawbacks that this work intends to overcome. First, scale-and-translate strategies to motion transferring in the image domain are not powerful enough to ensure the constraint of the motion. Second, learning-based techniques using 2D representation limit the method to be applied in actors that share the same skeleton proportions and that were captured from similar view angles. Because of the problems described above, we estimate the human pose in 3D and employ a volumetric representation of the human body, which is more suitable to capture body shape features.

We divided our motion estimation description into three different subsections. In the first subsection, we detail our representations of the human body and motion. In the second subsection, we describe the estimation of the 3D pose for each time instant. Then in the third subsection, we consider a set of consecutive frames together and regularize the motion in time.

#### 3.2.1.1 Human Body and Motion Representation

The most common model to represent human motion consists of two parts. One part details the skeleton's hierarchy and initial pose, and the second part describes the operation that takes the initial pose to the desired pose for each frame. The skeleton is defined by a kinematic tree of a set of joints. The kinematic tree consists of the initial location of the root joint, offsets of each joint from their parent, and rotational parameters for each joint that represents the relative rotation of the joint with its parent.

The human skeleton is a complex system composed of many limbs and joints where these limbs follow specific rules of proportions. Thus, to capture the statistics of shape variation and limb-length proportions, we represent the structure of the skeleton together with the 3D shape of the human body using the Skinned Multi-Person Linear (SMPL) model [Loper et al., 2015] that represents a wide variety of body shapes in natural human poses. The SMPL model is a skinned vertex-based model, where a mean template mesh of $N = 6890$ vertices is controlled by two sets of parameters, one for body shape, the other for the pose. The initial mesh and pose are shown in Figure 3.2.

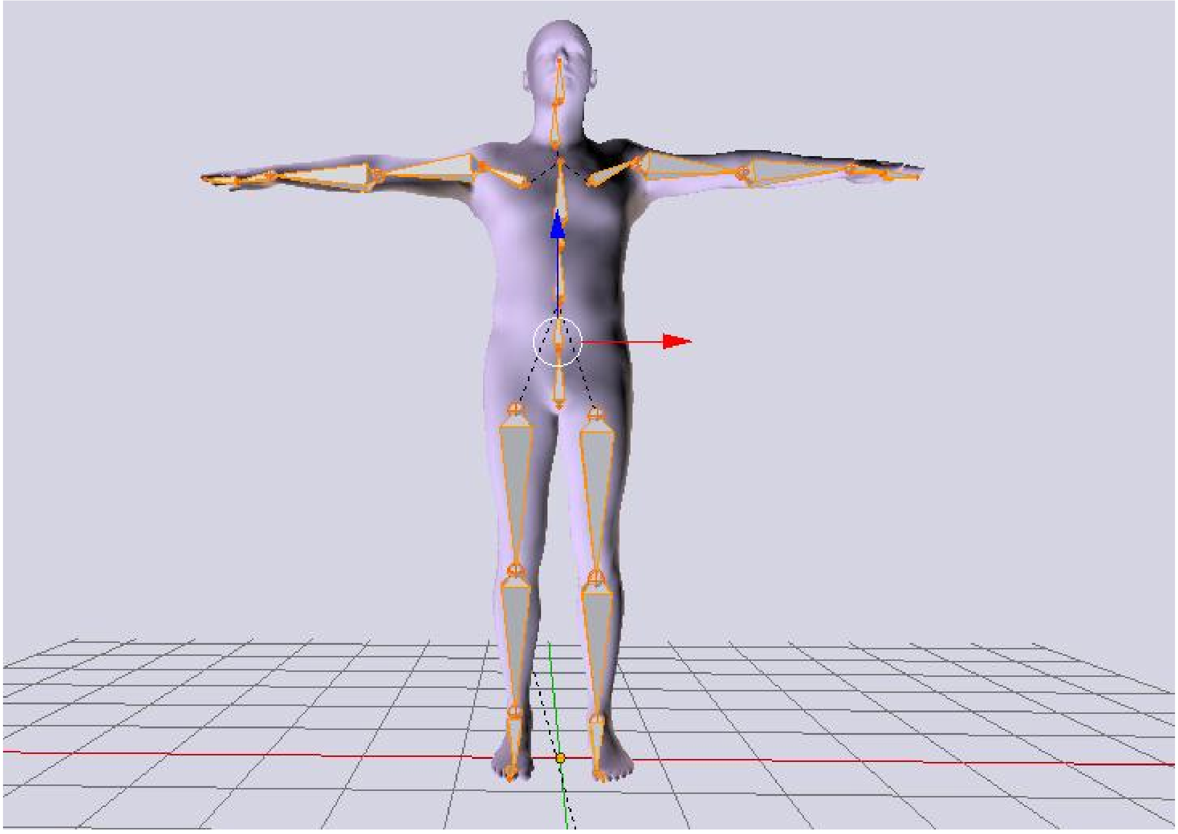

Figure 3.2: The initial mesh and pose in our motion representation using the SMPL model.

More formally, SMPL is defined as $M(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathcal{M})$, where $\boldsymbol{\beta}$ is a vector of shape parameters that are responsible for the 3D body shape due to identity, *i.e.*, how individuals vary in height, weight, and body proportions. The $\boldsymbol{\theta}$ is a vector of pose parameters representing body part rotations in a kinematic tree. The fixed parameters $\mathcal{M}$ represents the orthonormal principal components of shape displacements and the blend skinning weights that were learned from a large number of 3D body meshes.

Considering the SMPL shape coefficients $\boldsymbol{\beta} \in \mathbb{R}^{10}$ as our representation of hierarchy and initial pose, our motion is a set of translations and rotations of the skeleton joints over time. Therefore, we define $\mathbf{P}^i \in \mathbb{SE}(3)$ as the pose of a joint $i$. Each pose $\mathbf{P}^i$ is given by recursively rotating the joints of the skeleton tree, starting from the root joint and ending in its leaf joints, *i.e.*, the Forward kinematics (FK).

#### 3.2.1.2 Human Pose Model Fitting

In this section, we describe the estimation of the 3D pose for each frame. Our method builds upon the learning-based SMPL human pose/shape estimation framework of Kolotouros et al. [2019], whose objective is to infer the 3D human body and the

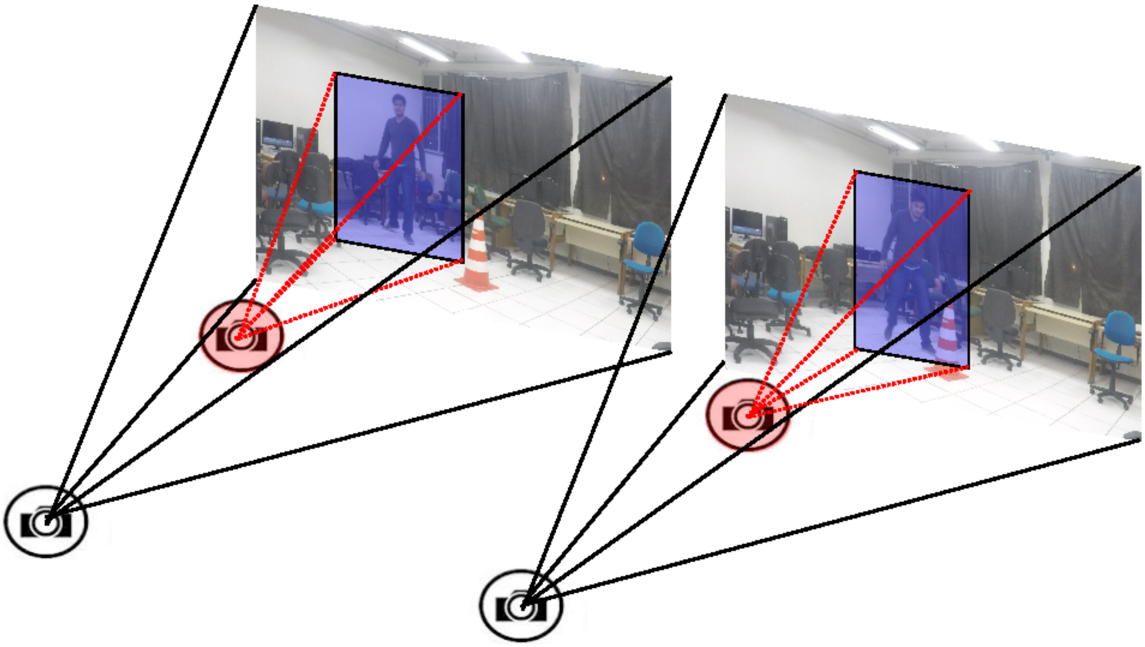

Figure 3.3: Example of global pose reconstruction. The input images cropping (blue), the resulting change of field of view (red), and the global camera coordinates (black).

camera pose such that its 3D joints project onto the annotated 2D joints. At the same time, an adversarial discriminator network is used to determine if the 3D parameters are real meshes from the unpaired data or not.

Kolotouros et al. [2019] predict pose in the coordinate system of the bounding box crop, where a weak-perspective camera model is adopted. The total number of parameters that represents the 3D reconstruction of a human body for each frame $k$ is a 85 dimensional vector composed of 10 shape coefficients ($\boldsymbol{\beta_k}$), 72 joint angles ($\boldsymbol{\theta_k}$), translation in axis $u$ and $v$ ($t_k \in \mathbb{R}^2$), and the scale ($s_k \in \mathbb{R}$).

According to Mehta et al. [2017], the bounding box cropping normalizes person in size and position, which frees 3D pose regression from having to localize the person in scale and image space. However, this strategy loses the global pose information, which is required to our motion transfer, see the Figure 3.3. Thus, after cropping the person using Openpose [Cao et al., 2017; Simon et al., 2017; Wei et al., 2016] and estimating the parameters that represents the 3D reconstruction, we map the reconstruction of Kolotouros et al. [2019] from the virtual camera coordinates to the original camera by minimizing an objective function that is the sum of two terms: one term that encourages the projections of the joints to remain in same locations into the global reference, and one term that encourages to keep the joints' angles. Together with this process, we force the subject shape to have same mean shape coefficients ($\boldsymbol{\beta^s}$) of the video. Thus, our energy function is given by:

$$E(\boldsymbol{\theta}_k, \mathbf{t}) = \lambda_1 E_J(\boldsymbol{\beta^s}, \boldsymbol{\theta}_k, \mathbf{t}, \mathbf{K}, \mathbf{J}_{2D}) + \lambda_2 E_{\boldsymbol{\theta}}(\boldsymbol{\theta}^s_k, \boldsymbol{\theta}_k), \tag{3.1}$$

where $\mathbf{t} \in \mathbb{R}^3$ is the translation, $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the camera intrinsic matrix, $\mathbf{J}_{2D}$ is the projections of the joints in the reconstruction of Kolotouros et al. [2019], and $\lambda_1$, $\lambda_2$ are scaling weights. Finally, the human model pose in each frame is then obtained with the forward kinematics (FK) in the skeleton tree:

$$\begin{pmatrix}\mathbf{P}^k_0 \; \mathbf{P}^k_1 \; \dots \; \mathbf{P}^k_{23}\end{pmatrix} = \text{FK}(\mathcal{M}, \boldsymbol{\beta^s}, \boldsymbol{\theta_k}), \tag{3.2}$$

where $\mathbf{P}^k_i$ is the pose of the joint $i^{th}$ or $\mathbf{P}^k_i = [\text{FK}(\mathcal{M}, \boldsymbol{\beta^s}, \boldsymbol{\theta_k})]_i)$. Consequently, considering the new set of $\boldsymbol{\theta}^s$ as the reconstructed set of $\boldsymbol{\theta}$, the raw actor motion is defined as $\mathbf{M}(\boldsymbol{\beta}^s, \boldsymbol{\theta}^s) = [\mathbf{P}_1 \; \mathbf{P}_2 \; ... \; \mathbf{P}_n] \in \mathbb{R}^{24\times4\times4\times n}$, where $\boldsymbol{\beta} = [\boldsymbol{\beta}^s_1, \boldsymbol{\beta}^s_2, \dots, \boldsymbol{\beta}^s_n] \in \mathbb{R}^{10\times n}$ and $\boldsymbol{\theta} = [\boldsymbol{\theta}^s_1, \boldsymbol{\theta}^s_2, \dots, \boldsymbol{\theta}^s_n] \in \mathbb{R}^{72\times n}$ are composed of the stacked $\boldsymbol{\beta}^s, \boldsymbol{\theta}^s_k$ over time.

#### 3.2.1.3 Motion Regularization

Since we estimate the character poses frame-by-frame, the resulting motion might present shaking motion with high-frequency artifacts in some short sections of the video. To alleviate these effects, we perform a regularization to seek a new set of joint angles $\widehat{\boldsymbol{\theta}}^s$ that creates a smoother motion. After applying a cubic-spline interpolation [De Boor et al., 1978] over the joints' motion $\mathbf{M}(\boldsymbol{\beta}^s, \boldsymbol{\theta}^s)$, we remove the outlier joints from the interpolated spline. The final motion estimate is obtained by minimizing the cost:

$$\min \Big( ||\widehat{\boldsymbol{\theta}}^s - \Theta||_2 + \gamma ||\text{FK}(\boldsymbol{\beta}^s, \widehat{\boldsymbol{\theta}}^s) - \mathbf{P}_{sp}||_2 \Big), \tag{3.3}$$

where $\Theta$ is the subset of inlier joints, FK is the forward kinematics, $\boldsymbol{\beta^s}$ defines the proportions and dimensions of the human body in the source video, $\mathbf{P}_{sp}$ is the spline interpolated joint positions, and $\gamma$ is the scaling factor between the original joint angles and the interpolated positions. This strategy removes high-frequency artifacts of the joints' motion while retaining the movement features.

Finally, we consider the final set of joint angles of the source video $\boldsymbol{\theta}^s$ as the set of joint angles $\widehat{\boldsymbol{\theta}}^s$ that creates a smoother motion.

### 3.2.2 Motion Retargeting

After estimating the motion from the input video, *i.e.*, $\mathbf{M}(\boldsymbol{\beta}^s, \boldsymbol{\theta}^s)$, and 3D model $\boldsymbol{\beta}^t$ of the target human, we can proceed to the motion retargeting step. Our second shared component (*Motion Retargeting*) is essential to guarantee that some physical restrictions are still valid during the target character animation. In this dissertation, we assume that the target character has a homeomorphic skeleton structure to the source character, *i.e.*, the geometric differences are in terms of bone lengths and body proportions. Our retargeting motion estimation loss is designed to guarantee the motion similarity and physical human-object interaction constraints over time. Similar to Gleicher [1998], our first goal is to retain the joint configuration of the target as close as possible to the source joint configurations at instant $k$, $\boldsymbol{\theta}^t_k \approx \boldsymbol{\theta}^s_k$, *i.e.*, to keep $\mathbf{e}_k$ small such as: $\boldsymbol{\theta}^t_k = \boldsymbol{\theta}^s_k + \mathbf{e}_k$. We also aim to keep similar movement style and speed in the retargeted motion. Thus, we propose a one step speed prediction in 3D space defined as $\Delta\mathbf{M}(\boldsymbol{\beta}, \boldsymbol{\theta}_k) = \mathrm{FK}(\boldsymbol{\beta}, \boldsymbol{\theta}_{k+1}) - \mathrm{FK}(\boldsymbol{\beta}, \boldsymbol{\theta}_k)$ to maintain the motion style from the original joints' motion:

$$\mathcal{L}_P(\mathbf{e}) = \sum_{k=i+1}^{i+n} ||\Delta\mathbf{M}(\boldsymbol{\beta}^t, \boldsymbol{\theta}^s_k + \mathbf{e}_k) - \Delta\mathbf{M}(\boldsymbol{\beta}^s, \boldsymbol{\theta}^s_k)||_1, \tag{3.4}$$

where $\mathbf{e} = [\mathbf{e}_{i+1}, \ldots, \mathbf{e}_{i+n}]^T$, and $n$ is the number of frames considered in the retargeting.

Rather than considering a loss for the total number of frames, we use only the frames belonging to a neighboring temporal window of $n$ frames equivalent to two seconds of video. This neighboring temporal window scheme allows us to track the local temporal motion style producing a motion that tends to be natural compared with a realistic-looking of the estimated source motion. The retargeting considering a local neighboring window of frames also results in a more efficient optimization.

#### 3.2.2.1 2D/3D human-to-object interactions

The human-to-object interactions (*i.e.*, motion constraints) are important to identify key features of the original motion that must be preserved in the retargeted motion. The specification of these interactions typically involves only a small amount of work in comparison with the task of creating new motions. Typical interactions are, for instance, that the target character's feet should be on the floor; holding hands while dancing or while grabbing/manipulating an object in the source video. Some examples of human-to-object motion constraints are shown in Figures 3.4 and 3.5, where the character is placing his left hand over a cone object and interacting with a box.

Going one step further than classic retargeting constraints defined in Gleicher [1998] and Choi and Ko [2000], where end-effectors must be at solely a desired 3D position at a given moment, we propose an extended hybrid constraint in the image domain, *i.e.*, the joint of the character must also be projected at a specific location in the

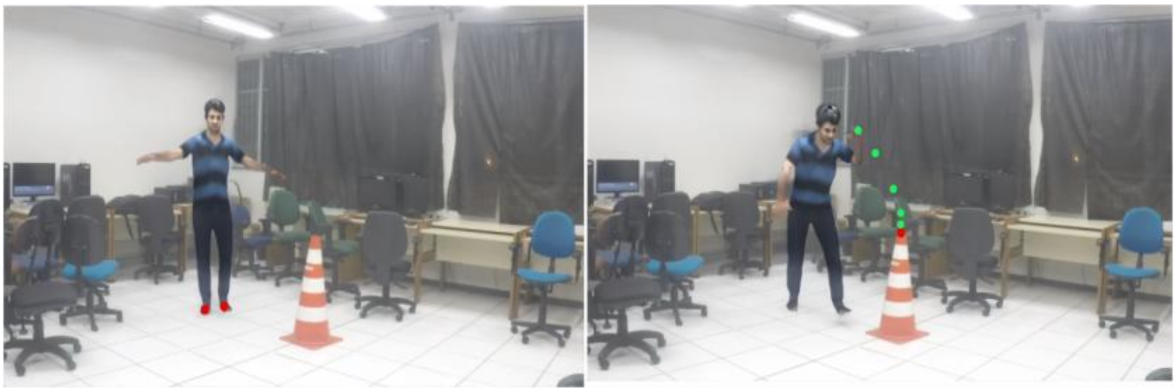

Figure 3.4: Example of 3D constraints from human-to-object interactions. The character is forced to use the original foot positions (red dots in left) and placing his left hand over a cone object (red dots in right). The retargeting preserves smoothness (green dots), *i.e.*, it does not include undesired high frequencies to the original motion.

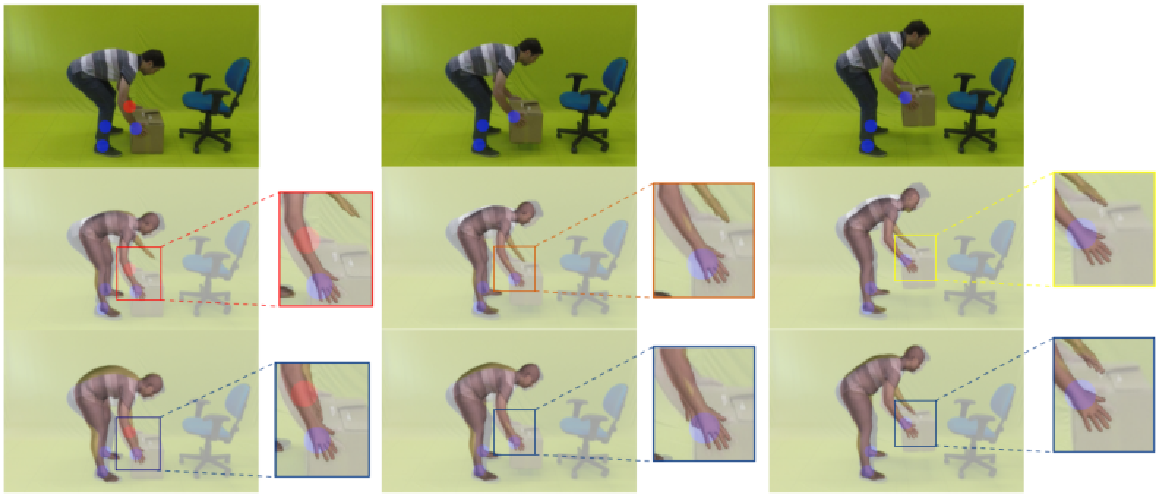

Figure 3.5: Example of hybrid 2D/3D constraints from human-to-object interactions. *Top row:* Original video with 3D constraints (blue dots) and 2D constraints (red dots). *Middle row:* Motion retargeting to a new character using only 3D constraints. *Bottom row:* The results for the new character applying hybrid 2D/3D constraints using our retargeting approach. Observe that the hands' positions are more consistent when adopting our hybrid strategy.

image. This type of motion constraint allows the user to exploit the visual knowledge of interactions of the actor in the scene. Some examples are shown in Figure 3.5, where two types of constraints are defined: 3D interactions (blue dots) impose the feet and right hand to be in the same location after the retargeting, and 2D constraints (red dots) imposing the correct position to the left hand in the image.

Our retargeting is capable of adapting to such situations by defining the motion retargeting constraints losses in respect to end-effectors' (hands, feet) 3D poses $\mathbf{P}_{R3D}$

and 2D poses $\mathbf{P}_{R2D}$ as:

$$\mathcal{L}_{R3D}(\mathbf{e}_k) = ||\mathrm{FK}(\boldsymbol{\beta}^t, \boldsymbol{\theta}_k^s + \mathbf{e}_k) - \mathbf{P}_{R3D}||_1, \tag{3.5}$$

$$\mathcal{L}_{R2D}(\mathbf{e}_k) = ||\Pi(\mathrm{FK}(\boldsymbol{\beta}^t, \boldsymbol{\theta}_k^s + \mathbf{e}_k), \mathbf{K}) - \mathbf{P}_{R2D}||_1. \tag{3.6}$$

where the $\Pi(., \mathbf{K})$ operator performs the projection taking a 3D point $(x, y, z)$ and projecting it into the image plane given the camera parameters $\mathbf{K}$.

#### 3.2.2.2 Space-time loss optimization

The final motion retargeting loss $\mathcal{L}$ combines the source motion appearance with the different shape and constraints of the target character from Equations 3.4, 3.5, and 3.6:

$$\mathcal{L} = ||\mathbf{W}_1\mathbf{e}||_2 + \lambda_1\mathcal{L}_P(\mathbf{e}) + \lambda_2\mathcal{L}_{R3D}(\mathbf{e}) + \lambda_3\mathcal{L}_{R2D}(\mathbf{e}), \tag{3.7}$$

where the joint parameters to be optimized are $\mathbf{e} = [\mathbf{e}_{i+1}, \dots, \mathbf{e}_{i+n}]^T$, $n$ is the number of frames considered in the retargeting window, $\lambda_1$, $\lambda_2$, and $\lambda_3$ are the contributions for the different error terms, and $\mathbf{W}_1$ is a positive diagonal matrix of weights for the motion appearance for each body joint. This weight matrix is set to penalize more errors in joints that are closer to the root joint.

One representative example of the retargeting strategy considering these hybrid 2D/3D constraints is shown in Figure 3.5. In this video sequence, the target actor is bigger and taller than the one in the source video (shown in the two first rows of the figure). Notice that the retargeting of the target actor (shown in the third row) results in more bent poses to maintain the human-to-object interactions, and thus the hands' positions are consistent when adopting our strategy.

### 3.2.3 Compositing

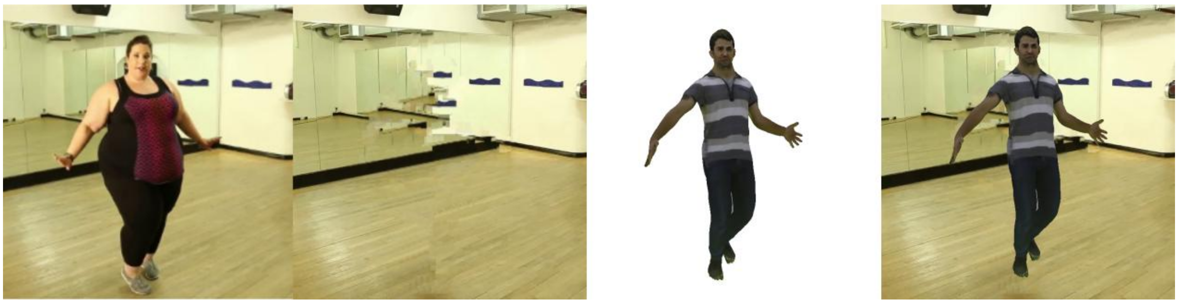

Figure 3.6: Example of the intermediate results of the compositing step. From left to right: Original image, background, rendering of the character and composition.

The third shared component is to compose the final image with the transferred person and the source background. We first segment the source image into a background layer using as a mask the projection of our computed model with a dilation. Next, the background is filled with the method proposed by Criminisi et al. [2004] to ensure temporal smoothness to the final inpainting. We compute the final pixel color value as the median value between the neighboring frames. Finally, the background and the target character are combined in the retargeted frame. Figure 3.6 shows an example of the intermediate results of the compositing step.

We also tested different inpainting formulations, such as the deep learning-based methods presented in Wang et al. [2019]; Yu et al. [2018]. Our experiments show that although these deep learning-based methods synthesize plausible pixels for each frame, the adopted inpainting strategy has better results considering the visual and spatio-temporal consistency between the frames.

## 3.3 Method I: Image-Based Rendering

2D human neural rendering approaches such as Wang et al. [2018], Chan et al. [2019], Aberman et al. [2018], and Sun et al. [2020], appeared as effective approaches for human appearance synthesis. However, these methods still suffer in creating fine texture details, notably in some body parts as the face and hands. Besides, it is well known that these methods suffer from quality instability when applied in contexts slightly different from the original ones, *i.e.*, a small difference in camera position, uncommon motions, pose translation, *etc.* These limitations motivate the proposal of our *Image-Based Rendering* method, which is designed to leverage visibility map information and semantic body parts to refine the initial target mesh model while keeping finer texture details in the transferring.

For a recap, the proposed methodologies to transfer jointly human motion and appearance are based on four components introduced in Section 3.1. Thus, we first estimate the motion of the character performing actions in the source video using the shared component proposed in Section 3.2.1. Second, we extract the body shape and texture of the target character in the second video. We also extract the texture, refine body geometry and estimate the visibility information of the body parts for transferring the appearance. Then, the shared *retargeting* component (see Section 3.2.2) adapts the estimated movement to the body shape of the target character while considering temporal motion consistency and the physical human interactions (constraints) with the environment. Finally, the *image-based rendering and*

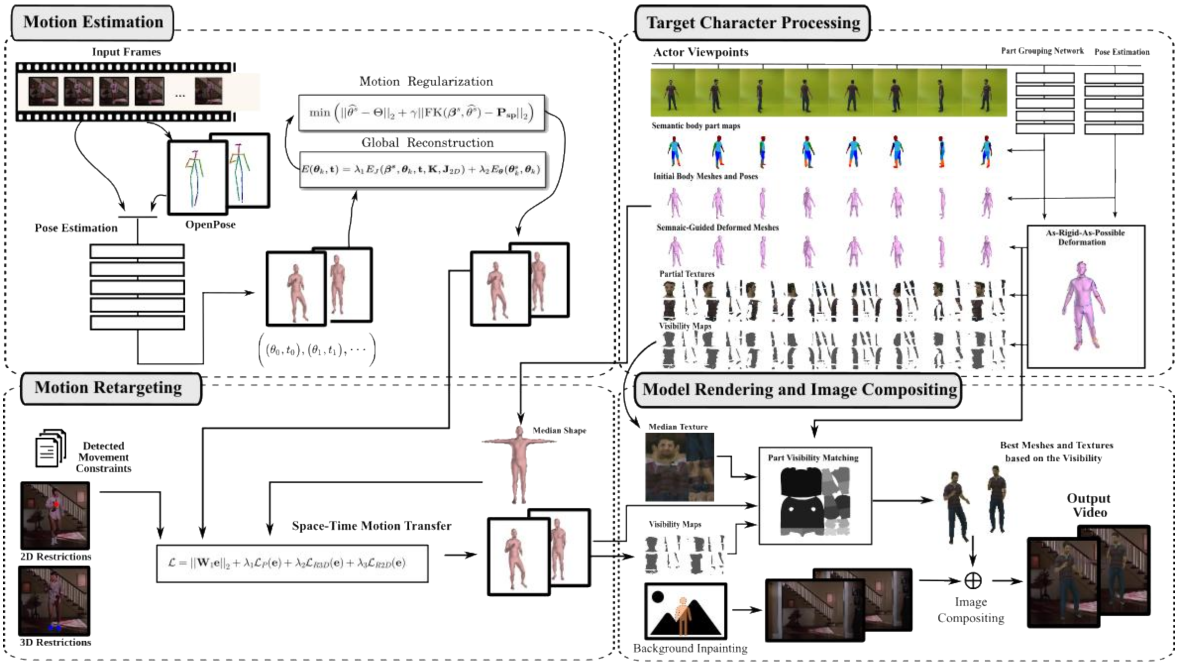
Motion Estimation
Input Frames
...
OpenPose
Pose Estimation
Motion Regularization
$\min\left(\|\widehat{\theta}^{s}-\Theta\|_2+\gamma\|\mathrm{FK}(\boldsymbol{\beta}^{s},\widehat{\theta}^{s})-\mathbf{P}_{\mathbf{sp}}\|_2\right)$
Global Reconstruction
$E(\boldsymbol{\theta}_k,\mathbf{t})=\lambda_1 E_J(\boldsymbol{\beta}^{s},\boldsymbol{\theta}_k,\mathbf{t},\mathbf{K},\mathbf{J}_{2D})+\lambda_2 E_{\boldsymbol{\theta}}(\boldsymbol{\theta}_k^{s},\boldsymbol{\theta}_k)$
$\left((\theta_0,t_0),(\theta_1,t_1),\cdots\right)$
Target Character Processing
Actor Viewpoints
Part Grouping Network
Pose Estimation
Semantic body part maps
Initial Body Meshes and Poses
Semnaic-Guided Deformed Meshes
As-Rigid-As-Possible Deformation
Partial Textures
Visibility Maps
Motion Retargeting
Detected Movement Constraints
2D Restrictions
3D Restrictions
Median Shape
Space-Time Motion Transfer
$\mathcal{L}=\|\mathbf{W}_1\mathbf{e}\|_2+\lambda_1\mathcal{L}_P(\mathbf{e})+\lambda_2\mathcal{L}_{R3D}(\mathbf{e})+\lambda_3\mathcal{L}_{R2D}(\mathbf{e})$
Model Rendering and Image Compositing
Median Texture
Part Visibility Matching
Visibility Maps
Best Meshes and Textures based on the Visibility
Output Video
Image Compositing
Background Inpainting

Figure 3.7: Overview of our Image-Based Rendering approach. Our method is composed of four main components: human motion estimation in the source video (first component); we retarget this motion into a different target character (second component), considering the motion constraints (third component), and by last, we synthesize the appearance of the target character into the source video.

*composition* component combines classical geometry rendering and image-based rendering to render the texture (appearance extracted from the target character) into the background of the source video. Figure 3.7 shows a schematic representation of our *Image-Based Rendering* method.

In the next subsection, we describe the *Target Character Processing* component of our *Image-Based Rendering* method, which is designed to leverage visibility map information and semantic body parts to refine the initial target mesh model while keeping finer texture details in the transferring. In the sequence, we present the improvements made to the *Compositing* step.

### 3.3.1 Target Character Processing

In order to create a more stable method and overcome the lack of details, we design a new semantic-guided image-based rendering approach that copies local patterns from input images to the correct position in the generated images. Our idea stems from using semantic information of the body (*e.g.*, face, arms, torso locations, *etc.*) in the geometric rendering to encode patch positions and image-based rendering to copy pixels from the target images, and therefore maintaining texture details. This strategy estimates a generic target body model $\boldsymbol{\beta}^t$, comprising body geometry, texture, and visibility information at each frame that will be transferred to the source video in the retargeting step.

#### 3.3.1.1 Semantic-Guided Human Model Extraction

When extracting the appearance and geometry of the human body, the self-occlusion of body parts and the deformable nature of the human body (and of clothes) bring challenging conditions. In order to tackle these difficulties, we propose a semantic-guided image-based rendering of body parts that explores the global and local information of the human body into the body model estimation.

While we gathered the global geometric information from the pose and shape as discussed in Section 3.2.1, the local geometric information is extracted for each viewpoint and aligned with the contours of their semantic body parts in the image. To perform this alignment, we partitioned and computed the correspondence of the 3D body model into fourteen meaningful body part labels (face, left arm, *etc.*). The 2D semantic body labels are computed using the Gong et al. [2018]'s body parsing model, which we fine-tuned using the people-snapshot dataset [Alldieck et al., 2018].

After computing the semantic map of the body, for each contour point (red squares in Figure 3.8) in the map, we define the vertex from the body mesh with the same semantic and the smallest Euclidean distance to the contour point as a control point (blue circles in Figure 3.8). The Euclidean distance is computed between the contour point and the 2D projection of the vertex. Each control point will receive a target position given its correspondent contour point. These control points and their new locations guide the deformation of the mesh to fit the shape into the semantic map's contour. In the deformation, we seek a new body mesh $Q$ that is a locally rigid transformation of the source body mesh $P$, following the control points given by the semantic contours. The mesh deformation is solved efficiently with the local rigid deformation As-Rigid-As-Possible (ARAP) [Levi and Gotsman, 2015]. The correspondence between each contour point and control point is represented in Figure 3.8 with colored small lines. Notice that the desired motion of the control points guides the deformation to fit the body mesh into the contours of the semantic map.

#### 3.3.1.2 Human Textures and Visibility Maps Extraction

The geometric information allows rendering the human target character in a new viewpoint by applying the desired transformations and re-projecting them onto the image

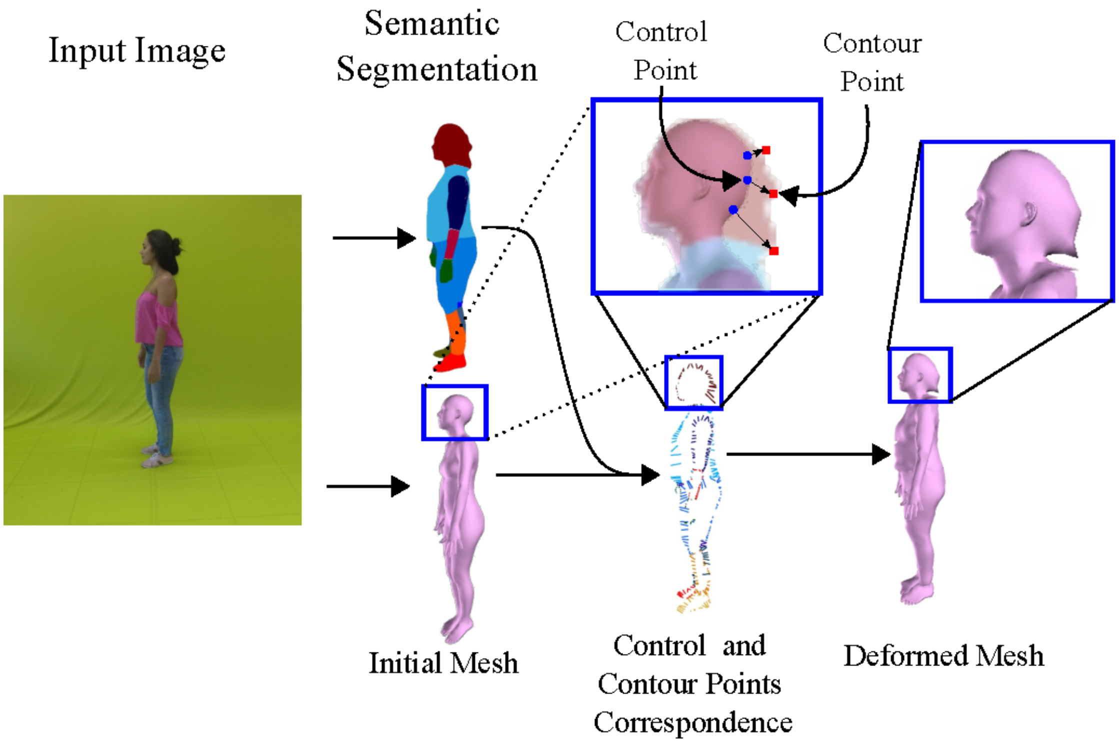

Figure 3.8: Semantic guided deformation. The contour points (red squares) in the semantic map indicate the target localization of the control points (blue circles) in the body mesh that it will guide the ARAP algorithm to fit the body mesh into the contours of the semantic map. The correspondence between each contour point and control point is illustrated with colored small lines.

plane. In order to compare and merge the information from the human actor from different viewpoints, we map the views to a common UV texture map space. The common UV texture map space is depicted in Figure 3.9. The mapping function is given by a parametric function $\mathcal{S}$ that maps a point in the mesh from frame $k$ to a point in the texture space with coordinates $(u, v)$, $\mathcal{S} : \mathbb{R}^3 \to \mathbb{R}^2$. Then, the accumulated texture map $UV(u, v)$ for all available images of the target character is done by the rendering with $\mathcal{S}$:

$$UV(\mathcal{S}(x, y, z)) = \mathcal{I}(\Pi((x, y, z), \mathbf{K})), \tag{3.8}$$

where the $\Pi(., \mathbf{K})$ operator performs the rendering taking a 3D mesh point $(x, y, z)$ and projecting it into the image plane given the camera parameters $\mathbf{K}$, and $\mathcal{I}$ is the texture information in the image coordinates.

Finally, we assert which mesh points are visible by exploring the inverse map $\mathcal{S}^{-1}(u, v)$, as illustrated in Figure 3.10. Each visibility map indicates which parts of the body model are visible per frame. Then we select the closest viewpoint to the desired new viewpoint, for each part of the body model from the visibility maps.

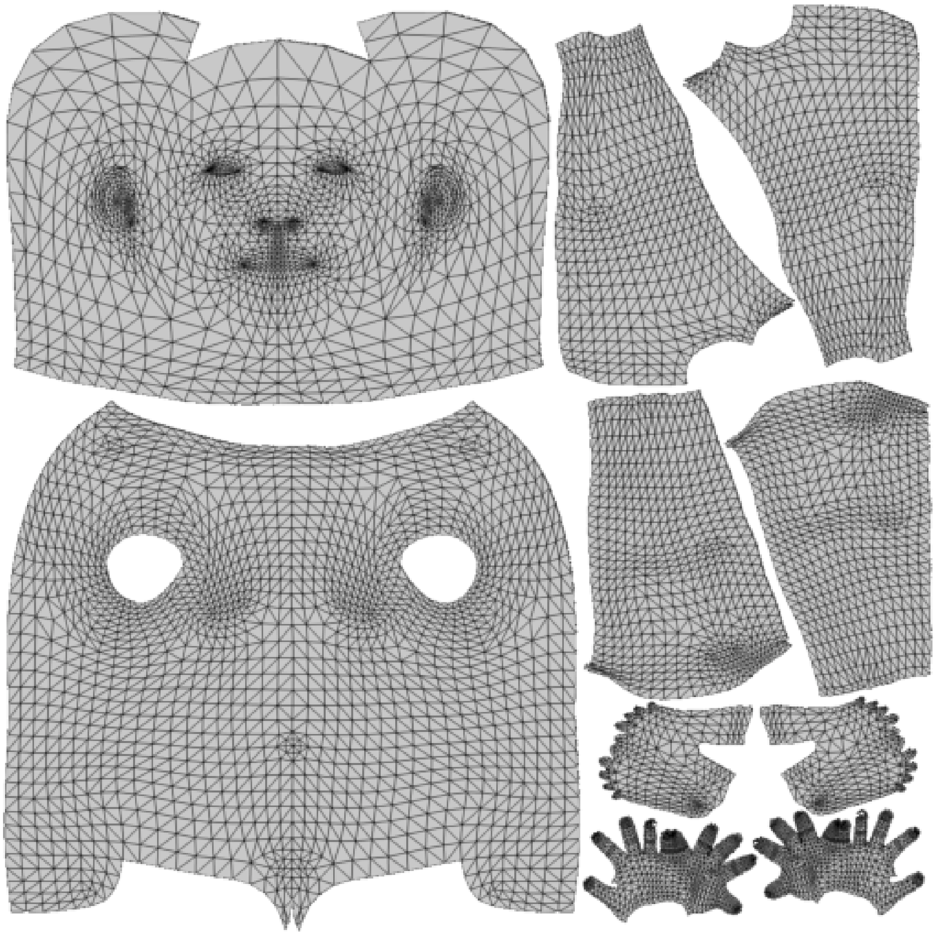

Figure 3.9: Our common UV texture map space. Each vertex in the mesh is mapped to a specific point in the texture space.

### 3.3.2 Model Rendering and Image Compositing

In our framework's last step, we combine the rendered target character and the source background to make a convincing final image. We explored the visibility map of the retargeted body model (global geometric information discussed in Section 3.3.1.2) to select the human body parts that better matches the target parts. Since the transformation between the retargeted SMPL model and the estimated SMPL model to the images is known, we apply the same transformation used in the local geometric information to move them to the correct positions. Instead of directly applying 3D warping in the selected images, we use our pre-warping step (texture map) to improve the rendering speed of 3D warping.

In order to fill the remaining part holes in the warped image, we explored the median of the accumulated texture $UV$ (see Figure 3.10). Finally, the background and the target character are combined in the retargeted frame.

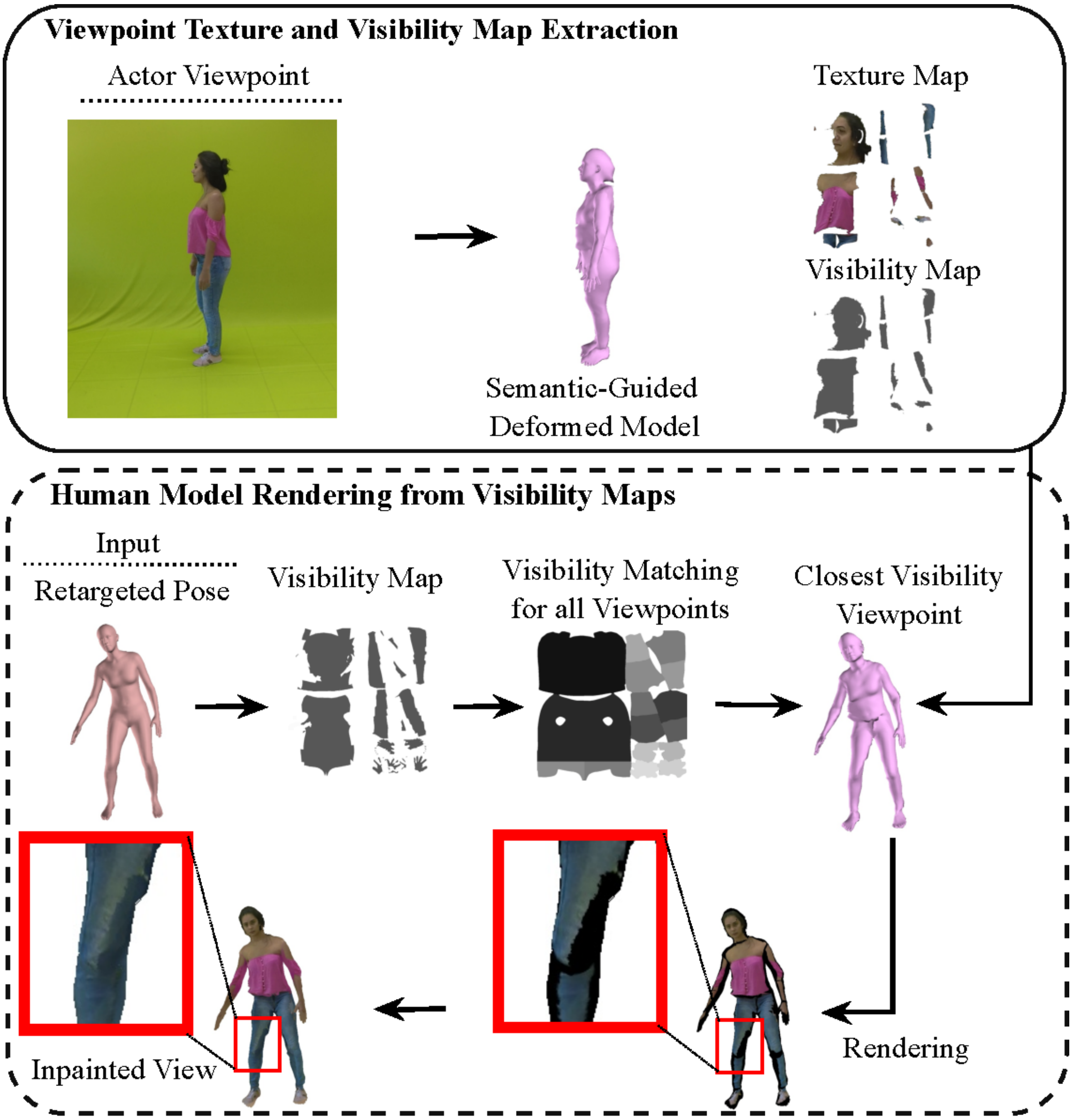
Viewpoint Texture and Visibility Map Extraction
Actor Viewpoint
Texture Map
Visibility Map
Semantic-Guided Deformed Model
Human Model Rendering from Visibility Maps
Input
Retargeted Pose
Visibility Map
Visibility Matching for all Viewpoints
Closest Visibility Viewpoint
Rendering
Inpainted View

Figure 3.10: Rendering of the visibility maps and texture images. *Top:* We project each target actor viewpoint in a common UV texture space using the estimated geometry and create a binary map of visibility body parts. *Bottom:* Given the goal pose (retargeted pose), we estimate its visibility body parts map, and then select the better matching visibility body parts created from the viewpoints from the target actor.

## 3.4 Method II: 3D Differentiable Human Rendering

Image-based rendering techniques like our previous technique are effective solutions to create 3D texture-mapped models of people, capable of synthesizing images from any arbitrary viewpoint without using a large number of images. On the other hand, image-based rendering methods cannot improve the visual quality of the synthesized images by using more data when available. Furthermore, the deformation process proposed in our previous method is not fast enough to be used in real-time applications. Thus, we offer a strategy to take advantage of all available data and, in addition, reduce inference time at the cost of increasing preprocessing time (training time).

Therefore, we propose a data-driven method, *i.e.*, a method that requires training a different model for each target person to perform the transferring. This training is computationally intensive, but the inference is very fast. Furthermore, we take a step towards combining learning and image-based rendering approaches in a new end-to-end architecture that synthesizes human avatars capturing both body geometry and texture details. The proposed architecture comprises a graph convolution network (GCN) operating over the mesh with a differentiable rendering approach [Liu et al., 2019b]. This architecture estimates a refined mesh and a detailed texture map to properly represent the person's shape and appearance for a given set of input frames.

Our method is also composed of four main steps: i) After estimating the motion in the source (see Section 3.2.1); ii) we extract the 3D shape and appearance from the target person using an end-to-end approach; iii) Then, by adapting the motion to constraints such as the different body proportions and physical constraints (see Section 3.2.2); iv) At last, the texture (appearance) extracted from the target is mapped into the 3D shape and is rendered in the source video (see Section 3.2.3). In the next subsection, we describe our end-to-end *Target Character Processing.*

### 3.4.1 Target Character Processing

In order to generate a deformable 3D texture-mapped human model, our end-to-end architecture has three main components to be trained during the rendering. The first component models local deformations on the human 3D body shape extracted from the images using a three-stage GCN. In the second component, a CNN is trained to estimate the human appearance map. Similar to the GCN, the CNN is also trained in a self-supervised regime using the gradient signals from the differentiable renderer. Finally, the third component comprises an adversarial regularization in the human appearance texture domain to ensure the reconstruction of photo-realistic images of people. In the inference/test time, we can feed our architecture with generic meshes parametrized by the SMPL model, and then we can create a refined mesh and a detailed texture map to represent the person's shape and appearance properly. Figure 3.11 outlines these components and their relations during the training phase and Figure 1.4 outlines these components during the inference phase.

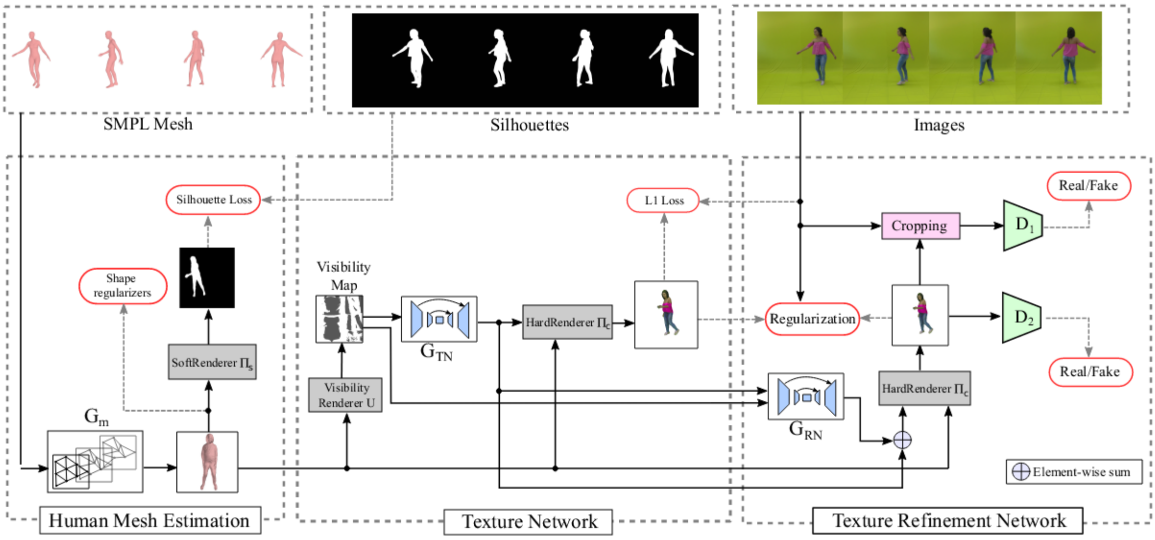

Figure 3.11: 3D differentiable human rendering strategy. Our architecture has three main networks: **a)** the *Human Mesh Estimation* that comprises a three-stage GCN and learns to fit and deform the mesh based on rendered silhouettes and shape regularizers; **b)** the *Texture Network*, a CNN that is trained conditioned on the visibility map generated from the deformed mesh to create a coarse texture by rendering and optimizing on the $l_1$ norm; **c)** the *Texture Refinement Network*, a second CNN that is conditioned on the visibility map and the coarse texture. It is trained in an adversarial manner to generate high frequency detail textures by rendering. A regularization loss based on the $l_1$ norm is also applied to the rendered images to prevent artifacts.

#### 3.4.1.1 Mesh Refinement Network

After computing the global human shape and pose information $P = \text{SMPL}(\boldsymbol{\theta}, \boldsymbol{\beta})$, we model local deformations on the mesh with the GCN Mesh Network $G_m$. Since the SMPL parametrization only provides a coarse 3D shape and cannot accurately model fine structures like clothes, we feed the mesh network with the initial SMPL mesh to refine its vertex positions with a sequence of refinement stages. The network is designed to learn the set of offsets to the SMPL mesh to increase the realism of the generated views. Drawing inspiration from the network architecture of Gkioxari et al. [2019], our refinement network is composed of three blocks with six graph convolutional layers with intermediate features of size 128. Each block in our mesh network performs four operations:

- *Vertex normal computation.* This operation computes the normal surface vector for each vertex as being the weighted sum of the normals of faces containing the vertex, where the face areas are used as the weights. The resulting normal is assigned as the node feature $f_i$ for the vertex $v_i$ in the GCN;

- *Graph convolution.* This convolutional layer propagates information along mesh edges using a aggregation strategy. Similar to Gkioxari et al. [2019], given the input vertex feature $f_i$, the layer updates the feature as $f_i' = \mathrm{ReLU}(W_0 f_i + \sum_{j \in \mathcal{N}} W_1 f_j)$, where $W_0$ and $W_1$ are learned weighting matrices, and $\mathcal{N}(i)$ gives the i-th vertex's neighbors in the mesh;
- *Vertex refinement.* To improve the quality of the vertex position estimation, this operation computes vertex offsets as $u_i = \tanh(W[f_i'; f_i])$. $W$ is a learned weighting matrix;
- *Vertex refinement clamping.* To avoid strong deformations (large $||u_i||_2$), we constraint and bound the position update of each vertex $v_i$ as $v_i' = v_i + \min(\max(u_i, -K(v_i)), K(v_i))$, where $K(v)$ is the 3D update bounds allowed to the vertex $v$, depending on the body part it belongs to. Each body part, *e.g.*, face, footprints, hands, head, torso, arms, feet, *etc.*, have predefined bound thresholds. This operation ensures that the offsets do not exceed the threshold defined to that body part, and that the refinement of the mesh geometry do not affect the body's topology.

Each of the three stages of the mesh network produces an intermediate mesh that is further refined by the next stage.

**Loss function.** For learning the mesh refinement, our model exploits two differentiable renderers that emulate the image formation process. Techniques such as presented by Liu et al. [2019b]; Ravi et al. [2020] enable us to invert such renderers and take advantage of the learning paradigm to infer shape and texture information from the 2D images.

During the training, the designed loss minimizes the differences between the image silhouette extracted from the real input image $I_s$ and the image silhouette $\hat{I}_s$ of the human body obtained by rendering the deformed 3D human model $M$ into the image by *SoftRenderer*, a differentiable renderer $\Pi_s(M)$. *SoftRenderer* is a differentiable that synthesises the silhouette of the actor (see images examples in Figure 3.12). We can define the loss of the Mesh Network $G_m$ as:

$$\mathcal{L}_m = \lambda_{gl} \mathcal{L}_{gl} + \lambda_{gn} \mathcal{L}_{gn} + \mathcal{L}_s, \tag{3.9}$$

where $\mathcal{L}_{gl}$ and $\mathcal{L}_{gn}$ regularize the Laplacian and the normals consistency of the mesh respectively Ravi et al. [2020], $\lambda_{gl}$ and $\lambda_{gn}$ are the weights for the geometrical regularizers,

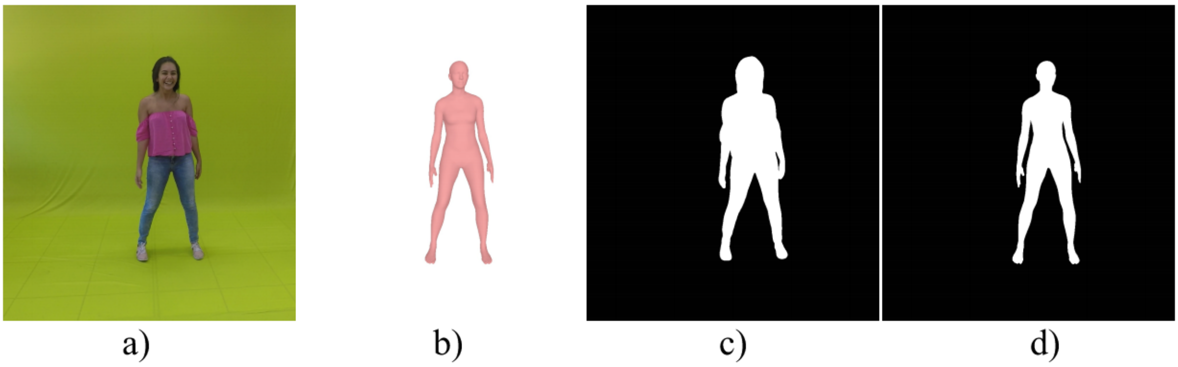

Figure 3.12: Images examples that compose our loss functions in method II : **a)** the real image $I$; **b)** the deformed 3D human model $M$; **c)** the image silhouette extracted from the input real image $I_s$, **d)** the image silhouette $\hat{I}_s$ of the human body obtained by rendering the deformed 3D human model.

and

$$\mathcal{L}_s = 1 - \frac{\left\|\hat{I}_s \otimes I_s\right\|_1}{\left\|(\hat{I}_s + I_s) - \hat{I}_s \otimes I_s\right\|_1} \tag{3.10}$$

is the silhouette loss proposed by Liu *et al.* Liu et al. [2019b], where $\hat{I}_s = \Pi_s(M)$, $M = G_m(P)$ is the refined body mesh model, and $\otimes$ is the element-wise product.

#### 3.4.1.2 Human Texture Generation

Similar to our previous method, we represent the appearance of the human model as a texture map in a common UV space (see Figure 3.9) that is applied to the refined mesh, in our case, the SMPL mesh with offsets. Our entire pipeline for texture generation is depicted in Figure 3.11-b-c and consists of two stages: coarse texture generation and then texture refinement. In an initial stage, given the refined 3D meshes of the actor $M$, the Texture Network $G_{TN}$ learns to render the appearance of the actor in the image. This texture is also further used to condition and regularize the Texture Refinement Network $G_{RN}$in the second stage.

**Texture Network.** We start estimating a coarse texture map with a U-Net architecture [Isola et al., 2017], as shown in Figure 3.13. The input of the network is a visibility map $x_v$ and it outputs the texture map $x_p$. The visibility map indicates which parts in the texture map must be generated to produce a realistic appearance for the refined mesh. The visibility map is extracted by a parametric function $x_v = U(M)$ that maps points of refined mesh $M$ with positive dot product between the normal vector and the camera direction vector to a point $x_v$ in the texture space. We implement $U$ as a render

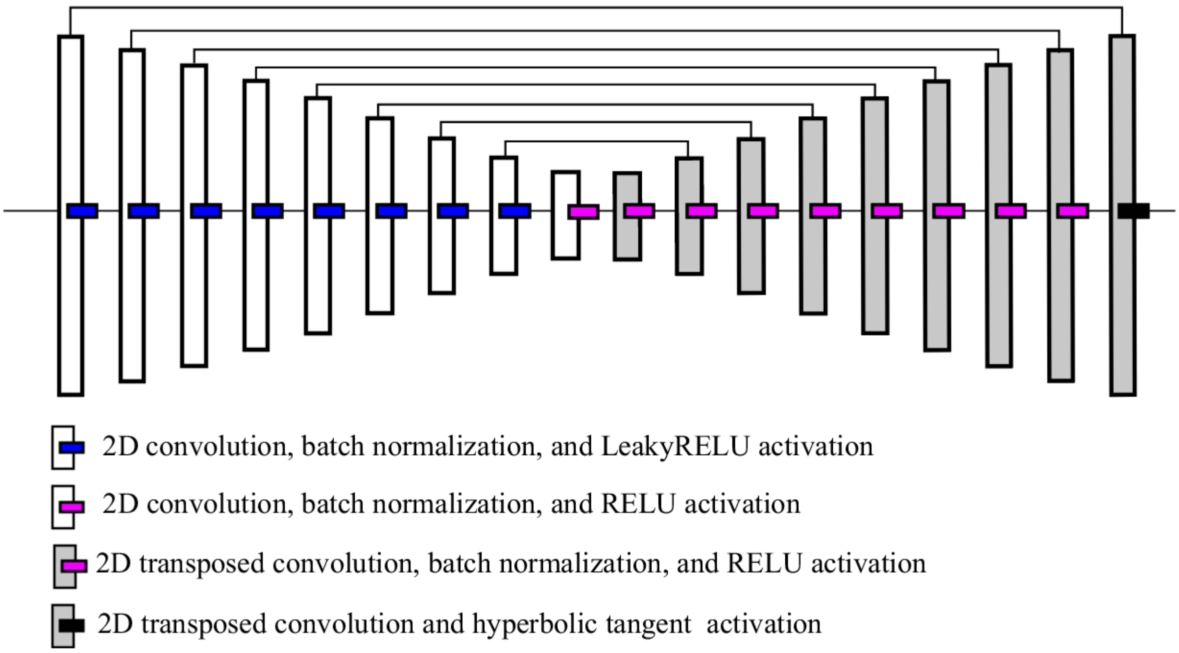
2D convolution, batch normalization, and LeakyRELU activation
2D convolution, batch normalization, and RELU activation
2D transposed convolution, batch normalization, and RELU activation
2D transposed convolution and hyperbolic tangent activation

Figure 3.13: Our U-net architecture. The U-Net is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

of the 2D UV-map considering only faces with positive dot product between the normal vector and the camera direction vector. Thus, the network can learn to synthesize texture maps on-demand focusing on the important parts for each viewpoint.

The Figure 3.11-b shows a schematic representation of the Texture Network training. The *HardRenderer* $\Pi_c(M, x_p)$, represents the colored differentiable renderer that computes the output coarse textured image $\hat{I}$, of model $M$ and texture map $x_p$. In our case, $\hat{I} = \Pi_c(M, G_{TN}(U(M))$. Conversely to the *SoftRenderer*, the differentiable *HardRenderer* is used to propagate the detailed human texture appearance information (color). Specifically, we train the Texture Network to learn to generate a coarse texture by imposing the $l_1$ norm in the person's body region of the color rendered image as:

$$\mathcal{L}_{pt} = \left\|(\hat{I} - I) \otimes B\right\|_1 / \|B\|_1, \tag{3.11}$$

where $I$ is the real image in the training set and $B$ is the union of visibility masks and real body regions.

**Texture refinement.** We further improve the coarse texture to represent finer details. For that, we design the Texture Refinement Network $G_{RN}$ to condition the generation of a new texture map from the coarse texture, on a coherent output of the Texture Network $G_{TN}$ and the visibility map. Our Texture Refinement Network has the same

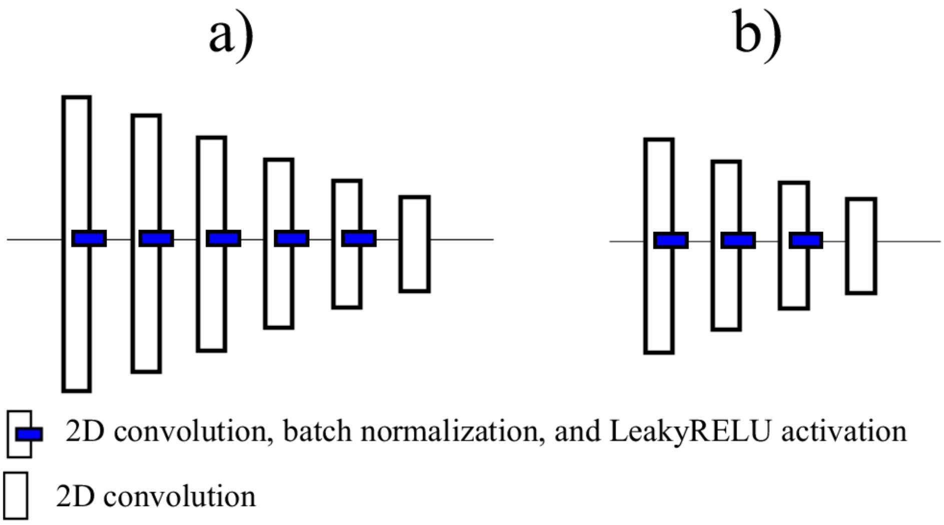
a)
b)
2D convolution, batch normalization, and LeakyRELU activation
2D convolution

Figure 3.14: Schematic representation of our discriminators' architecture: the Image Discriminator (**a**) and Face Discriminator (**b**).

architecture as our Texture Network, as shown in Figure 3.13.

In our adversarial training, the Texture Refinement Network acts as the generator network $G_{RN}$ and engages in a minimax game against two task-specific discriminators: the Face Discriminator $D_1$ and Image Discriminator $D_2$. The generator is trained to synthesize texture maps in order to fool the discriminators which must discern between "real" images and "fake" images, where "fake" image are produced by the neural renderer using the 3D texture-mapped model estimated by the Mesh and Texture Networks. While the discriminator $D_1$ sees only the face region, the Image Discriminator $D_2$ sees the whole image. Figure 3.14 depicts a schematic representation of our discriminators' architecture.

These three networks are trained simultaneously and drive each other to improve their inference capabilities, as illustrated in Figure 3.11-c. The Texture Refinement Network learns to synthesize a more detailed texture map to deceive the discriminators which in turn learn differences between generated outputs and ground truth data. The total loss function for the generator and discriminators for the rendering is then composed of three terms:

$$\begin{gathered}\min_{G_{RN}}(\max_{D_1}\mathcal{L}_{GAN}(G_{RN}, D_1)+\\ \max_{D_2}\mathcal{L}_{GAN}(G_{RN}, D_2) + \mathcal{L}_r(G_{RN})),\end{gathered} \tag{3.12}$$

where both $\mathcal{L}_{GAN}$ terms address the discriminators and $\mathcal{L}_r$ is a regularization loss to reduce the effects from outlier poses. Each adversarial loss is designed as follows:

$$\begin{aligned} \mathcal{L}_{GAN}(G, D) = \mathbb{E}_y[\log D(y)] + \\ \mathbb{E}_{x_v,x_p}[\log(1 - D(\Pi_c(M, G(x_v, x_p))))], \end{aligned} \tag{3.13}$$

where $x_v$ is the visibility map, $x_p$ is the output of the Texture Network, $M$ is the refined mesh, and $y$ is the corresponding segmented real image $I \otimes B$.

Finally, to reduce the effects of wrong poses, which causes mismatches between the rendered actor silhouette and silhouette of the real actor, we also add a regularization loss to prevent the GAN from applying the background's color into the human texture. The first term of the regularization loss acts as a reconstruction of the pixels by imposing the $l_1$ norm in the person's body region, and the second term enforces eventual misaligned regions to stay close to the coarse texture:

$$\begin{aligned} \mathcal{L}_r = \alpha_1 \Big\|(I - \hat{I}^{RN}) \otimes B\Big\|_1 / \|B\|_1 + \\ \alpha_2 \Big\|(\hat{I}^{TN} - \hat{I}^{RN}) \otimes C\Big\|_1 / \|C\|_1, \end{aligned} \tag{3.14}$$

where $\alpha_1$ and $\alpha_2$ are the weights, $\hat{I}^{TN}$ is the rendered image using the coarse texture, $\hat{I}^{RN}$ is the rendered image using the refined texture, and $C$ is the misaligned regions without the face region, *i.e.*, the image region where the predicted silhouette and estimated silhouette are different.

## 3.5 Summary and Closing Remarks

In this chapter, we presented two human transferring methods that simultaneously consider body shape, motion retargeting constraints, and human-to-object interactions over time, while retaining visual appearance quality. In method I, our carefully designed approach based on a rendering pipeline reduces the effects of poor generalization to changes in camera viewpoint, scale, and camera intrinsics and allows the proposed method to be applied even if a few images are available. In method II, our data-driven approach takes advantage of both differentiable rendering and the 3D parametric model to produce a fully 3D controllable human model using all available data. In addition, this approach reduces the inference time at the cost of increasing preprocessing time (training time).

# Chapter 4

# Human Retargeting Dataset and Evaluation

Datasets are one cornerstone of recent advances in Computer Vision. While there are many large datasets available for human shape and pose estimation [Andriluka et al., 2014; Lin et al., 2014; Mahmood et al., 2019], existing motion retargeting datasets are yet rare, hampering progress on this area. Due to the lack of suitable benchmark datasets, recent works on neural view synthesis and body reenacting [Chan et al., 2019; Esser et al., 2018; Wang et al., 2018] only analyze their results in qualitative terms or quantitative terms to self-transfer in which the source and target actors are the same subjects. The provided data is adapted to the requirements of their method setup, and it is hard to perform a comparison with other methods on this data. Cross-transfer is far more complicated than self-subject transfer. First, self-transfer is not affected by body appearance changes (from body shape, and clothing for example). Second, the self-shape transfer does not account for human-to-object interactions or disregards the influence of existing human-environment physical interactions.

## 4.1 Existing Datasets

Existing video retargeting datasets are still rare. Villegas et al. [2018] provided human joint poses from synthetic paired motions, however, paired visual information is not available, limiting the appearance transferring. Chan et al. [2019] made available videos with random actor movements that can be used to learn the appearance of the target actor. However, the provided data is limited to their setup requirements, and it does not allow the analysis and comparison with other methods. Liu et al. [2019] presented a set of videos with random actions of target subjects, as well as videos of the subjects performing an A-pose movement. This set enables methods focusing on modeling 3D human body estimation or using few keyframes to be executed using their data. On the flip side, the lack of paired motions limits motion and appearance retargeting results in quantitative terms, where the source and target actors are different subjects. Conversely, our proposed dataset, in addition to videos with random actions of the target subjects and videos of the subjects performing an A-pose video, also provides several carefully paired reconstructed 3D motions and annotated human-to-object interactions in the image and 3D space.

## 4.2 Human Retargeting Dataset

To evaluate the retargeting and appearance transfer with different actor motions, consistent reconstructed 3D motions, and with human-to-object interactions, we created a new dataset with *paired motion* sequences from different characters and *annotated motion retargeting constraints*. For each video sequence, we provide a refined 3D actor reconstructed motion and the actor body shape estimated with Alldieck et al. [2018]. The refined reconstructed 3D motions and 2D-3D annotation of interactions were collected by manual annotation. The provided motions are not prone to typical motion artifacts such as bended knees, body shape variations, and camera-to-actor translation changes. Figure 4.1 shows the annotated labels.

Our carefully designed dataset comprises eight subjects. To keep the dataset with diversity, we choose participants (subjects S0 to S7) with different gender, sizes, clothing styles, and body shapes. Figure 4.2 shows the subjects, their respective body models and labels. A short description of these subjects is as follows:

- S0: A female character, 1.65 meters height, and neat hourglass body shape;
- S1: A male character, 1.85 meters height, and inverted triangle body shape;
- S2: A male character, 1.70 meters height, and rectangle body shape;
- S3: A male character, 1.83 meters height, and pear body shape;
- S4: A male character, 1.83 meters height, and lean column body shape;
- S5: A male character, 1.81 meters height, and apple body shape;
- S6: A male character, 1.84 meters height, and rectangle body shape;
- S7: A female character, 1.54 meters height, and pear body shape.

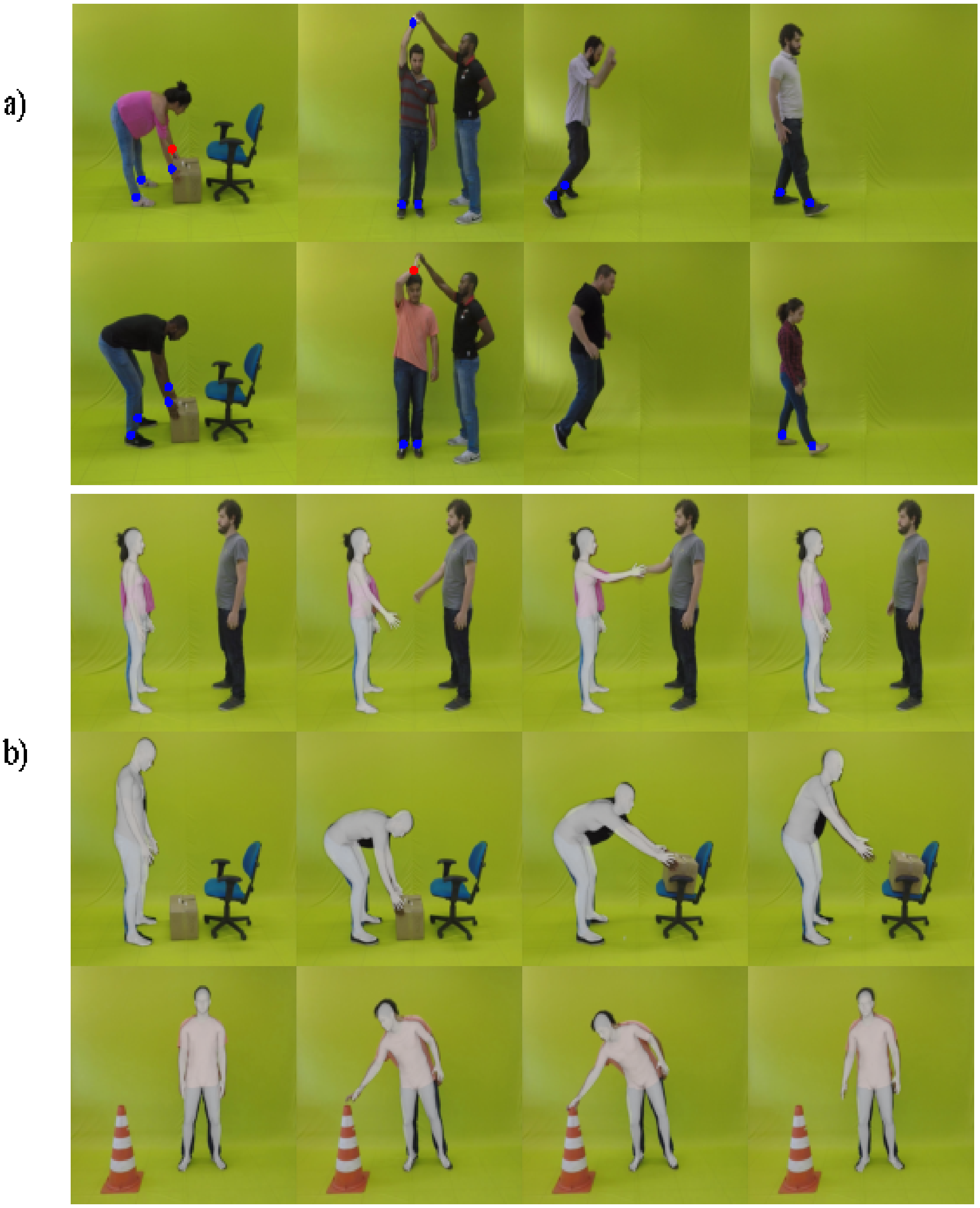

Figure 4.1: Human retargeting dataset. *a)* Paired motions (upper and lower rows) with annotated motion constraints (3D constraints in blue and 2D constraints in red). *b)* The reconstructed 3D motions.

We also selected a set of movements that are representative to the problem of the retargeting, increasing level of difficulty and with different motion constraints.

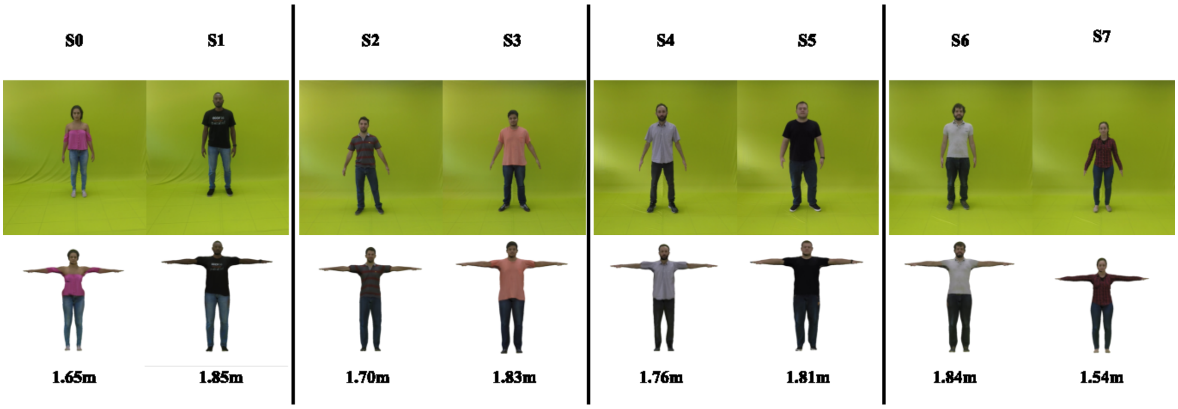

Figure 4.2: The subjects participating in our dataset. *F*irst row: Subject number. *S*econd row: Subject's image. *T*hird row: Subject's body model. *L*ast row: Their respective height.

These movements are of "picking up a box", "spinning", "jumping", "walking", "shaking hands", "touching an object", "pulling down", and "fusion dance". Figure 4.3 shows each movement performed by a different subject. A brief description of the recorded videos and the motion restrictions presented therein is as follows:

- **Jump:** The subjects were instructed to jump from a side of the scene to another, trying to jump the same distance at every jump. The restrictions are in frames where the feet touch the floor plane. In this motion, the subjects of different heights (their height often correlates to the length of legs) have difficulties making this motion because of the distance between each jump. For instance, the actors with lower height should adapt their pose to be able to reproduce the movement. These videos sequences have around 100 frames.

- **Walk:** The subjects walked from a side of the scene to another. The restrictions of this movement are when the subject touches the floor with his/her feet. In this motion, subjects with lower height should adapt their legs to be able to give a step with the same distance. These videos sequences have around 100 frames.

- **Spinning:** The subjects were instructed to rotate like a ballet dancer, while another person holds their right hand extended upwards. The restrictions of this motion are in the right hand of the subject touching the hand of the second person, and in the subject's feet when he/she is doing the spinning movement. These videos sequences have around 230 frames.

- **Shake Hands:** The subjects were instructed to stay in the same position, then performing a handshaking movement. The restrictions of this motion are in the

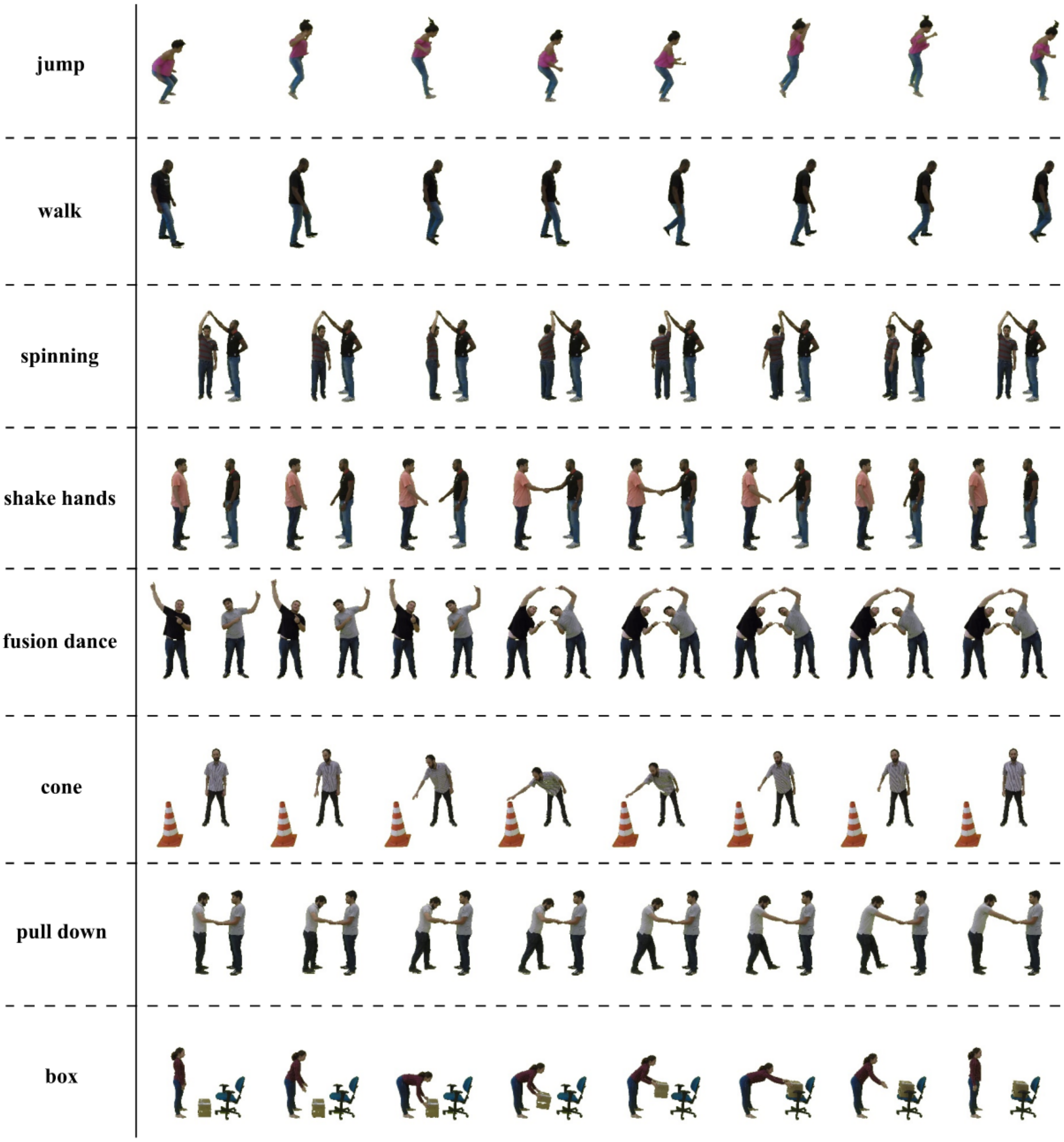

Figure 4.3: Overview of all motions and subjects presented in our proposed dataset.

feet of the subjects and in the subjects' hands when they are in contact. These videos sequences have around 110 frames.

- **Fusion Dance:** The subjects were instructed to perform a "fusion dance" like in the anime Dragon Ball Z. The restrictions of this motion are in the feet of the subjects and in the subjects' hands when they touch each other. These videos sequences have around 60 frames.
- **Cone:** The subjects were instructed to stay at a defined position, then move

their body without taking off the feet of the floor, and touch a cone placed in front of them. The restrictions in this motion are the subjects' feet touching the floor, and in frames where the hand of the subject touches the cone. In this motion, the subjects made different body poses because of the distance between them and the cone in the scene. These videos sequences have around 150 frames.

- **Pull Down:** The subjects were instructed to start in a defined position, then take a step backward to another defined position, while holding a hand of another person. The restrictions of these motions are the feet of the subject touching the floor plane and subject's hand touching the hand of the other person placed in the scene. These videos sequences have around 70 frames.
- **Box:** The subjects were instructed to hold a position, then pick up a box at the floor and put it on a chair placed in front of them, without taking off their feet of the floor. The restrictions of this motion are the subject's feet touching the floor, in the subject's hands when picking up the box and while moving the box to the chair. These videos sequences have around 210 frames.

Each actor performed all eight actions. We paired two actors to perform the same motion sequence, where the subjects were instructed to follow marks on the floor to perform the same action, resulting in four paired videos per action (a total of 32 paired videos). Then, we define the combination of actors aiming at the most challenging configuration for the task of human retargeting motion. For instance, actors $S0$, $S2$, $S4$, and $S6$ were paired respectively with $S1$, $S3$, $S5$, and $S7$. We also provide, for each subject, three videos: one video where the subject is rotating and holding an A-pose, a four-minute video where the subject is performing different poses, and a 15-second video where the subject is dancing. All videos were recorded with 1,000 × 1,080 of resolution [1] at 30 frames per second. This information allows training most existing approaches for evaluation.

## 4.3 Protocol

The evaluation often employed by works on character reenacting/synthesis [Aberman et al., 2018; Chan et al., 2019; Liu et al., 2019; Sun et al., 2020] consists in setting the source character equal to the target character. However, we argue that this protocol is used because of the absence of paired motion sequences, as also noted in Liu et al. [2019] and Sun et al. [2020]. While this protocol might be appropriate to assess new

[1] This resolution was adopted in previous methods in the literature.

synthesized view/pose in the same scene background, we state that it is not appropriate for evaluating the synthesis of new videos of people when taking into account motion constraints (*e.g.*, human-to-object interactions), distinct shapes and heights, and transferring them in different backgrounds where they were initially recorded.

Therefore, we run our evaluation protocol as follows: when evaluating a new synthesized video, we place the target actor performing a similar motion to the source actor (all physical constraints and the human-to-object interactions are taken into account). Then, we move the target actor to a different scenario configuration to perform the retargeting. If the retargeting is successfully executed, the method will place the target actor moving as the source actor into the source actor's scenario.

## 4.4 Evaluation Metrics

We measure the quality of the synthesized frames in terms of the following metrics: i) the structural similarity (SSIM) [Wang et al., 2004] that compares local patterns of pixel intensities normalized for luminance and contrast. SSIM assumes that human visual perception is highly adapted for extracting structural information from objects; ii) learned perceptual similarity (LPIPS) [Zhang et al., 2018a], which provides a deep neural learned similarity distance closer to human visual perception; iii) the mean squared error (MSE) that is computed by averaging the squared intensity differences between the pixels; and iv) Fréchet video distance (FVD) [Unterthiner et al., 2019], which was designed to capture the quality of the synthesized frames and their temporal coherence in videos. These are widely used perceptual distances to measure how similar the synthesized images are in a way that coincides with human judgment.

To properly evaluate the quality of retargeting, it is required a paired video where the target character is the same as the source video sequence. Collecting real paired motion sequences from different characters is a challenging task, even when the movement of the actors is carefully predefined and synchronized. For instance, each actor has a movement style that can result in videos with unsynchronized actions (as seen in the third column in Figure 4.1-a). Thus, to make possible the computation of quantitative metrics, we relax the assumption that the frame $k$ in the synthesized video must be the frame $k$ in the paired video by applying a small window of acceptance around $k$, *i.e.*, we evaluate the quality of one synthesized frame as a better answer between the frame and a small window of frames in the paired video. The window of acceptance $w$ is estimated for each pair of videos according to:

$$w(V_1, V_2) = \max(15, 2 \times (|len(V_1) - len(V_2)|), \tag{4.1}$$

where $len(V)$ is the number of frames in the video $V$. Equation 4.1 captures how much two videos are not synchronized allowing the synthesized frames to match with the paired video. The lower bound value of 15 frames was empirically selected by a visual analysis of our videos.

Aside from the image quality metrics, we also propose to evaluate the approaches with fake image detectors. Since our dataset provides paired motion sequences, we can evaluate the retargeted frames' quality and realism with an image forgery detector. We applied the detector in the generated and real frames from the sequence where the same subject performs the retargeted motion. The retargeted frames were generated using motion extracted from source videos whose subject performing the motion is different from the target subject.

We adopted the image forgery detection algorithm presented by Marra et al. [2020] to evaluate all methods in our experiments. The Marra et al.'s method evaluates images by detecting pixel artifacts, which is a common metric for detecting CNN-generated images. We remark that we tested different image forgery detection algorithms such as the method proposed by Wang et al. [2020b], but the results were inconclusive, which might be because of the high resolution of our images. Furthermore, our images are not only generated by CNN methods, and as the authors stated, their method performs at chance in visual fakes produced by image-based rendering or classical commercial rendering engines. Finally, we define the forgery performance as the difference between the probability of the paired real image and its respective synthetic frame of being classified as fake. This difference indicates how far a synthetic frame is from being recognized as fake with a similar result of a real image.

## 4.5 Videos for Qualitative Analysis

Although our dataset contains several videos, it is essential to evaluate the behavior of our methodology in less controlled data acquisition. Thus, we provide three videos acquired under uncontrolled conditions. In our experiments, we used these videos to analyze the effectiveness of our method with more complex motions, background, and varying illumination. A short description of these videos is as follows:

- **joao-pedro**: Video with moderate frequency motions and a static background, where a 1.80 meters height male character is walking and interacting with a cone in the floor. The motion constraints are in the feet, and the hand, where the person touches the top of the cone object;

- **tom-cruise**[2]: Video with moderate frequency motions and large translation in all directions, where a 1.70 meters height male character is pretending to sing while dancing. The annotated restrictions are in the dancer's feet;
- **bruno-mars**[3]: Video with high-frequency motion where a 1.65 meters height male character is dancing. The motion has strong occlusions in the arms and feet. The annotated restrictions are in the dancer's feet. Moreover, this sequence was also used by Chan et al. [2019] and became a famous sequence for appearance transferring.

[2] https://www.youtube.com/watch?v=IUj79ScZJTo

[3] https://www.youtube.com/watch?v=PMivT7MJ41M

# Chapter 5

# Experiments and Results

In this chapter, we describe a set of experiments to analyze the gain related to our techniques based on the compromise of three main aspects: appearance, motion, and body shape. First, we present our baselines. Second, we show that our *Image-Based Rendering* method exhibit a competitive quality compared to recent 2D neural rendering techniques, besides having several advantages in terms of control and ability to preserve motion features and body shape. Finally, we analyze our *3D Differentiable Human Rendering* method.

## 5.1 Comparison with Previous Approaches

We compare our methods against four recent representative methods with different assumptions, including V-Unet [Esser et al., 2018], vid2vid [Wang et al., 2018], EBDN [Chan et al., 2019], and iPER [Liu et al., 2019]. V-Unet is a notorious representative of image-to-image translation methods using conditional variational autoencoders to generate images based only on a 2D skeleton and an image from the target actor. Similar to V-Unet, iPER is not data-driven, but it is a generative model trained in an adversarial manner. Vid2vid and EBDN methods, for their turn, are data-driven methods, *i.e.*, they require training a GAN for several days over one video of the target subject in a large set of different poses.

## 5.2 Method I: Image-Based Rendering

### 5.2.1 Implementation details

In the motion estimation, we used $\lambda_1 = 10^{-6}$ and $\lambda_2 = 10^{-2}$. In the human motion estimation and retargeting steps, we used $\gamma = 10$, $\lambda_1 = 5$, $\lambda_2 = 1$, and $\lambda_3 = 1$. We minimize the retargeting loss function with Adam optimizer using 300 iterations, learning rate 0.01, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. To deform the target body model with the semantic contour control points, we employed the default parameters proposed by Levi and Gotsman [2015], and we fed the method with eight images taken from different viewpoints of the actor (see Figure 3.7). For a complete comparison, we also present qualitative results of the new frames from the state-of-the-art approaches by replacing the generated background regions from the methods by the source video background.

### 5.2.2 Processing Time

Although vid2vid and EBDN required a few seconds to generate a new frame, the training step of vid2vid spent approximately 10 days on an NVIDIA Titan XP GPU for each target subject, and to run the fine-tuning of the EBDN took approximately four days to complete all stages for each subject. On the other hand, our first approach does not need to be trained. The significant parts of our method's processing time are the retargeting optimization, the deformation, and the model rendering. On an Intel Core i7-7700 CPU and NVIDIA Titan XP GPU, the average run-time for one frame of retargeting optimization was about 1.2 seconds, including I/O. The deformation took around 720 seconds on eight frames. The model rendering, *i.e.*, selecting the best texture map considering the visibility map, warping the selected parts, and filling all the holes in the texture, took about 30 seconds per frame. Thus, the total processing time $t(N)$ in seconds to run our method on a video with $N$ frames with a resolution of 1,920 × 1,080 is approximately $t(N) = 1.2 \times N + 720 + 30 \times N$ seconds.

### 5.2.3 Ablation Study

To verify that the space-time motion transfer optimization, motion regularization, semantic-guided deformation, and visibility maps contribute to our approach's success in producing more realistic frames, we conducted several ablation analysis using the motion sequences from our dataset.

Table 5.1 shows the results of our ablation study in terms of MSE and FVD of five ablated versions of the Method I. We draw the following observations. First,

Table 5.1: **Ablation study**. Comparison of mean MSE and FVD for different ablated versions of our method on all motion types of our dataset (best in bold).

| **Method** | MSE↓ | FVD↓ |
|---|---|---|
| No motion regularization | 275.47 | 887.94 |
| No semantic guidance | 275.88 | 879.86 |
| No 2D/3D constraints | 285.33 | 859.67 |
| No use of visibility mask | 273.45 | 789.06 |
| Full method | **259.75** | **738.96** |

Table 5.2: **Quantitative ablation analysis of the retargeting with different pair of actors.** Table shows the error in pixels between the constraints position and the end-effectors location. This error indicates how far the end-effector is from the target position (better close to 0).

| | **Pair of actors** | | | | |
|---|---|---|---|---|---|
| **Method** | S0-S1 | S2-S3 | S4-S5 | S6-S7 | Avg. |
| Direct Transfer | 27.41 | 22.84 | 10.06 | 22.88 | 20.80 |
| Retargeting | 4.57 | 3.19 | 2.67 | 5.19 | **3.90** |

Table 5.3: **Quantitative ablation analysis of the retargeting with different motion types.** Table shows the error in pixels between the constraints position and the end-effectors location. This error indicates how far the end-effector is from the target position (better close to 0).

| **Method** | **Motion type** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | jump | walk | spinning | shake hands | cone | fusion dance | pull down | box | Avg. |
| Direct Transfer | 13.73 | 14.49 | 21.90 | 18.22 | 16.77 | 24.67 | 34.11 | 22.49 | 20.80 |
| Retargeting | 2.44 | 1.94 | 2.04 | 2.90 | 2.76 | 10.66 | 3.14 | 5.35 | **3.90** |

the best result is achieved when the full method is applied. Second, removing the 2D/3D constraints reduces the performance in terms of MSE. These constraints play a key role in the compliance of the poses to motion constraints. By removing them, large fragments from the background are computed as part of the retarget character body when computing MSE using the paired video, leading to the worst MSE value. Third, without the shape-aware regularization, which hinders the temporal coherence, the model presents the worst value of FVD. We can also see that after removing the semantic guidance, which decreases the quality of the texture applied onto the 3D model, the frames have more artifacts, and the model also performs poorly in terms of FVD.

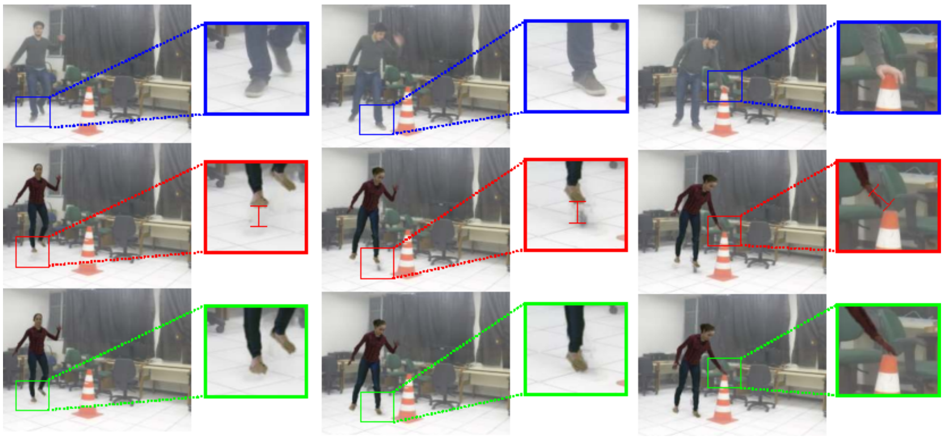

Figure 5.1: Qualitative ablation analysis of the retargeting. *Top row:* source video containing the actor S3. *Middle row:* transferring results to the actor S7 without the physical interactions. *Bottom row:* transferring results of our method with 2D-3D interactions.

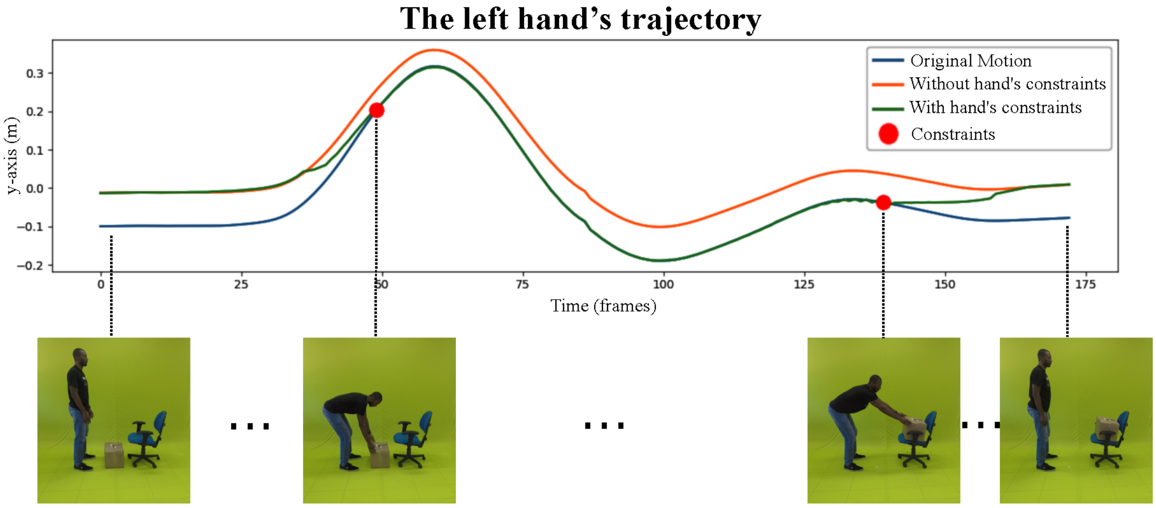

Figure 5.2: Retargeted trajectory with motion constraints. The curves show the left hand's trajectory on the y-axis when transferring the motion of *picking up a box* between two differently sized characters: original motion (blue line), a naïve transfer without constraints at the person's hand (red line), and with constraints (green line). Frames containing motion constraints are located between the red circles.

The results of a more detailed performance assessment of the effects from motion constraints in the video retargeting are shown in Table 5.2 and 5.3. The error between the computed position of the end-effectors and the target positions (motion constraints from the human-to-object interactions) is significantly smaller for all motion sequences and pairs of actors when applying our retargeting strategy. Some frames are shown in Figure 5.1 to illustrate the created visual artifacts when not considering the motion constraints. In this setup, the source actor (top row) is taller than the target (bottom row), and the target actor does not touch the floor nor touch with her hands the cone without the hybrid motion constraints (middle row). Conversely, these features are kept when considering the 2D/3D human-to-object losses in the retargeting. Another representative example of the retargeted motion trajectory over time, with one shorter actor interacting with a box, is shown in Figure 5.2. Please notice the smooth motion adaptation produced by the retargeting with the restrictions in frames 47 and 138 (green line) when the character's left hand is touching the box. Additionally, these results illustrate that our method is able to impose different space-time human-to-object interactions and motion constraints in the retargeting.

Table 5.4 shows the forgery performance when synthesizing the frames after removing the visibility map extraction and the semantic-guided human model extraction. We can see that these two steps significantly enhance the quality of the results to a point where the detector returns for the "shake hands" sequence a probability of 29.24 higher when removing these two components.

Table 5.4: **Visibility maps and semantic-guidance analysis**. Average forgery performance for each movement (better close to 0).

| Motion | Method | |
|---|---|---|
| | No Semantic-guidance and Visibility Map | Complete Model |
| jump | 46.97 | **31.11** |
| walk | 45.76 | **23.35** |
| spinning | 34.46 | **21.09** |
| shake hands | 34.91 | **5.67** |
| cone | 39.20 | **24.80** |
| fusion dance | 33.24 | **15.97** |
| pull down | 35.68 | **17.85** |
| box | 38.09 | **17.63** |

### 5.2.4 Quantitative Analysis

We performed the video retargeting in all sequences in our dataset, including several public dance videos also adopted in the works Chan et al. [2019] and Liu et al. [2019]. Table 5.5 shows the comparison of our approach in the dataset considering the motion types and pair of actors. We can see that despite not being dataset-specific, V-Unet and iPER did not perform well when the reference image is not from their datasets. One can see that our method outperforms, on average, all methods in considering SSIM, LPIPS, MSE, and FVD metrics. Regarding the experiments considering the pairs of actors, our first approach also achieved the best average results in all metrics. In particular,

Table 5.5: **Method I: Comparison with state of the art**. SSIM, LPIPS, MSE, and FVD comparison by motion types and pair of actors from our dataset (best in bold, second-best in italic).

| Metric | Method | Motion type | | | | | | | | | Pair of actors | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | jump | walk | spinning | shake hands | cone | fusion dance | pull down | box | Avg. | S0-S1 | S2-S3 | S4-S5 | S6-S7 | Avg. |
| SSIM↑ | V-Unet | 0.870 | 0.871 | 0.843 | 0.847 | 0.862 | 0.797 | 0.847 | 0.857 | 0.849 | 0.855 | 0.886 | 0.830 | 0.826 | 0.849 |
| | Vid2Vid | 0.880 | 0.884 | 0.856 | 0.858 | 0.878 | 0.821 | 0.859 | 0.866 | *0.862* | 0.868 | 0.901 | 0.848 | 0.835 | 0.862 |
| | EBDN | 0.878 | 0.880 | 0.855 | 0.859 | 0.878 | 0.820 | 0.857 | 0.858 | 0.861 | 0.867 | 0.898 | 0.844 | 0.834 | 0.861 |
| | iPER | 0.877 | 0.880 | 0.852 | 0.859 | 0.877 | 0.816 | 0.855 | 0.856 | 0.859 | 0.867 | 0.896 | 0.842 | 0.831 | 0.859 |
| | Method I (Ours) | 0.881 | 0.885 | 0.855 | 0.860 | 0.879 | 0.820 | 0.861 | 0.869 | **0.864** | 0.872 | 0.902 | 0.846 | 0.834 | **0.864** |
| LPIPS↓ | V-Unet | 0.147 | 0.132 | 0.157 | 0.161 | 0.174 | 0.243 | 0.166 | 0.158 | 0.167 | 0.184 | 0.160 | 0.166 | 0.158 | 0.167 |
| | Vid2Vid | 0.131 | 0.105 | 0.126 | 0.136 | 0.133 | 0.203 | 0.142 | 0.133 | *0.138* | 0.148 | 0.131 | 0.129 | 0.147 | 0.138 |
| | EBDN | 0.141 | 0.122 | 0.139 | 0.138 | 0.143 | 0.215 | 0.151 | 0.170 | 0.153 | 0.159 | 0.145 | 0.147 | 0.159 | 0.153 |
| | iPER | 0.151 | 0.134 | 0.151 | 0.151 | 0.155 | 0.239 | 0.168 | 0.184 | 0.167 | 0.161 | 0.165 | 0.170 | 0.171 | 0.167 |
| | Method I (Ours) | 0.125 | 0.099 | 0.130 | 0.131 | 0.128 | 0.206 | 0.131 | 0.127 | **0.135** | 0.133 | 0.129 | 0.133 | 0.143 | **0.135** |
| MSE↓ | V-Unet | 295.04 | 269.59 | 354.33 | 377.02 | 328.68 | 559.75 | 417.00 | 346.13 | 368.44 | 381.77 | 344.71 | 362.63 | 384.66 | 368.44 |
| | Vid2Vid | 257.33 | 206.42 | 286.18 | 332.09 | 253.04 | 452.77 | 349.28 | 288.54 | *303.32* | 312.40 | 274.80 | 269.81 | 356.26 | *303.32* |
| | EBDN | 306.92 | 269.58 | 312.69 | 312.12 | 266.17 | 463.79 | 384.23 | 361.57 | 334.63 | 324.40 | 314.10 | 331.83 | 368.19 | 334.63 |
| | iPER | 313.43 | 275.22 | 344.94 | 314.92 | 267.39 | 504.00 | 404.79 | 377.16 | 350.23 | 277.98 | 328.07 | 358.30 | 436.57 | 350.23 |
| | Method I (Ours) | 237.16 | 178.57 | 286.86 | 270.25 | 237.86 | 434.64 | 301.66 | 245.86 | **274.11** | 243.95 | 260.14 | 294.88 | 297.46 | **274.11** |
| FVD↓ | V-Unet | 1,491.63 | 845.44 | 1,721.81 | 1,257.20 | 1,415.24 | 1,712.93 | 2,437.98 | 1,816.94 | 1,587.39 | 2,239.14 | 1,352.10 | 1,856.78 | 1,108.34 | 1,639.09 |
| | Vid2Vid | 879.94 | 266.6 | 1,085.49 | 396.31 | 790.79 | 997.42 | 997.96 | 1,069.85 | 810.48 | 778.53 | 719.80 | 762.46 | 574.08 | *708.72* |
| | EBDN | 887.56 | 273.00 | 918.94 | 423.08 | 725.49 | 952.22 | 1,113.46 | 853.26 | *768.37* | 791.98 | 751.45 | 560.27 | 826.71 | 732.60 |
| | iPER | 1,770.31 | 656.07 | 1,531.64 | 1,266.14 | 1,051.42 | 1,322.72 | 1,440.94 | 1,719.55 | 1,344.84 | 1,270.41 | 1,092.82 | 1,395.81 | 1,214.64 | 1,243.42 |
| | Method I (Ours) | 1,119.50 | 330.91 | 674.99 | 478.93 | 767.68 | 791.01 | 988.35 | 760.33 | **738.96** | 715.00 | 653.30 | 720.49 | 515.61 | **651.10** |

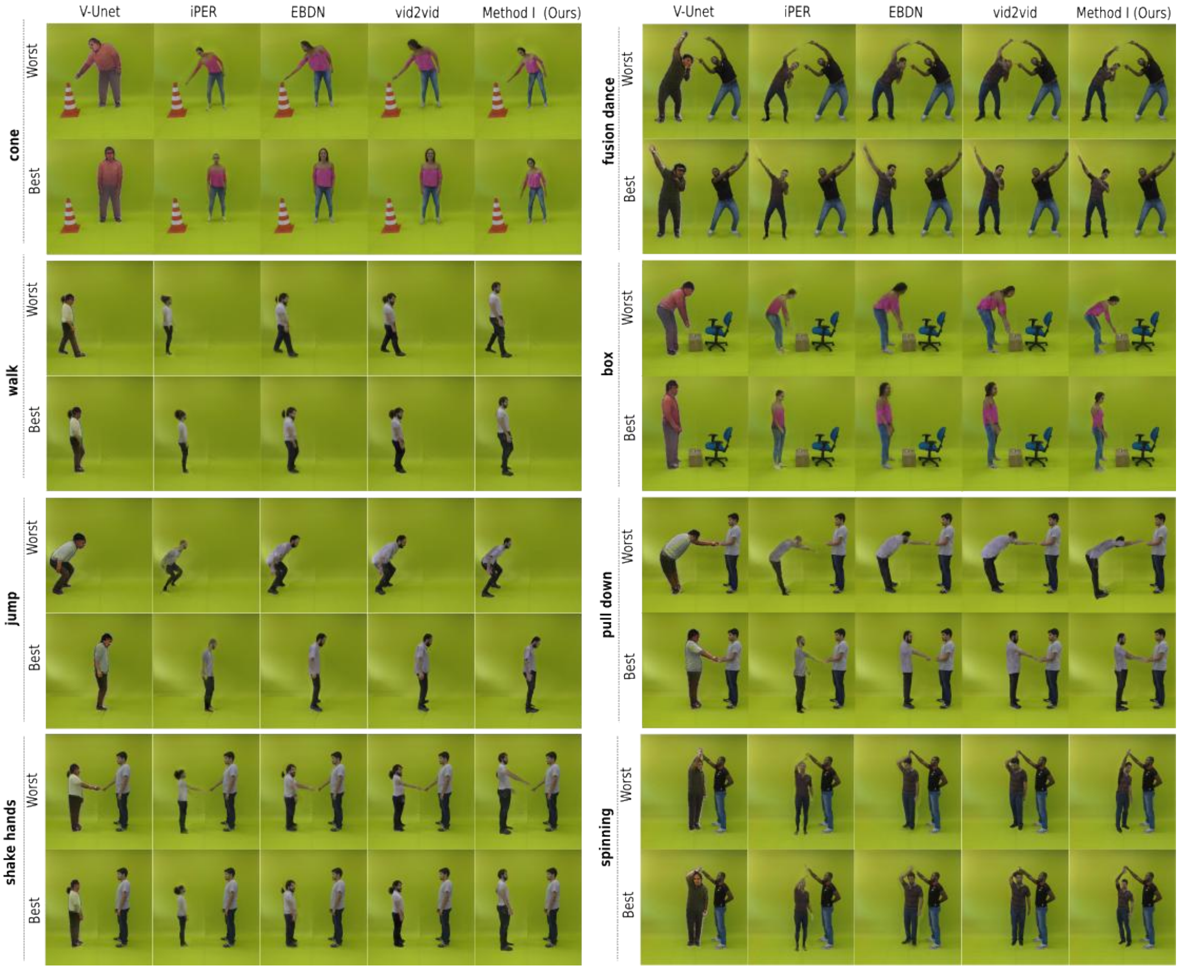

Figure 5.3: Method I: Qualitative analysis in the dataset sequences. Transferring results considering the cases where the person is not standing parallel to the image plane or has the arms in front of the face. In each sequence: the first row shows the worst generated frame for each method and the second row presents the best generated frame for each method.

our method presented better results when the subjects have different heights (S0-S1 and S6-S7). We ascribe this performance to our method being aware of the shape and physical interactions, which allows it to correct the person's position when the source actor is taller or smaller than the target person. Moreover, it is noteworthy that the sequences "spinning" and "fusion dance" are challenging for all methods, including our methodology that was affected by wrong pose estimations. Our first approach was slightly outperformed by vid2vid only in these two sequences.

Table 5.6 shows the results for the experiments on image forgery detection. These experiments indicate that the fake detector in the frames generated by our method has the closest performances to real images. For instance, in the "shake hands" sequence, the probability of a frame synthesized by our method to be fake is only 5.67% higher

Table 5.6: **Forgery performance**. Quantitative metrics of the movement transfer approaches considering the forgery performance metric (better close to 0).

| Motion | Method | | | | |
|---|---|---|---|---|---|
| | EBDN | iPER | vid2vid | V-Unet | Method I (Ours) |
| jump | 39.46 | 46.56 | 41.66 | 47.41 | **31.11** |
| walk | 41.00 | 44.27 | 30.79 | 45.79 | **23.35** |
| spinning | 33.64 | 33.56 | 32.36 | 34.51 | **21.09** |
| shake hands | 31.71 | 33.56 | 19.39 | 34.93 | **5.67** |
| cone | 38.23 | 37.84 | **24.74** | 39.25 | 24.80 |
| fusion dance | 26.07 | 32.64 | 16.49 | 33.38 | **15.97** |
| pull down | 31.30 | 34.65 | 19.85 | 35.67 | **17.85** |
| box | 35.96 | 37.11 | 24.70 | 38.12 | **17.63** |

than when applying the detector to the respective real frame.

### 5.2.5 Qualitative Analysis

We evaluated the capability of our method to transfer motion and appearance and retain interactions of the original motion despite the target actor having different proportions to the actor in the source video.

Some frames used to compute the metrics in Table 5.5 are shown in Figure 5.3. One can note the large discrepancy between the quality of the frames for the same method. As shown in Figures 5.3 and 5.4, the end-to-end learning techniques have impressive results when the person is standing parallel to the image plane and the arms are not in front of the face; however, these methods perform poorly when the person is out of these contexts, such as when bending. The proposed first method, for its turn, retains the same quality for most poses.

We also analyzed the impact of the camera pose and the actor's scale in the quality of the resulting videos of the methods. We transferred two target persons from our dataset to two videos with different camera setups and image resolutions: "bruno-mars" and "tom-cruise" sequences. In the "bruno-mars" sequence, shown in Figure 5.4, we ran vid2vid and EBDN using their respective solution to tackle with different image resolutions and actors with different proportions from the training data. Vid2vid's strategy (scaling to keep the aspect ratio and then cropping the image) results in an actor with different proportions compared with the training data, which leads to a degradation of quality. EBDN's strategy (pose normalization, scaling to keep the aspect ratio, and then cropping the image) keeps the similarity between the input video and training data, but body and face occlusions are still problems. The red squares highlight these issues on the right side in Figure 5.4, where EBDN and vid2vid

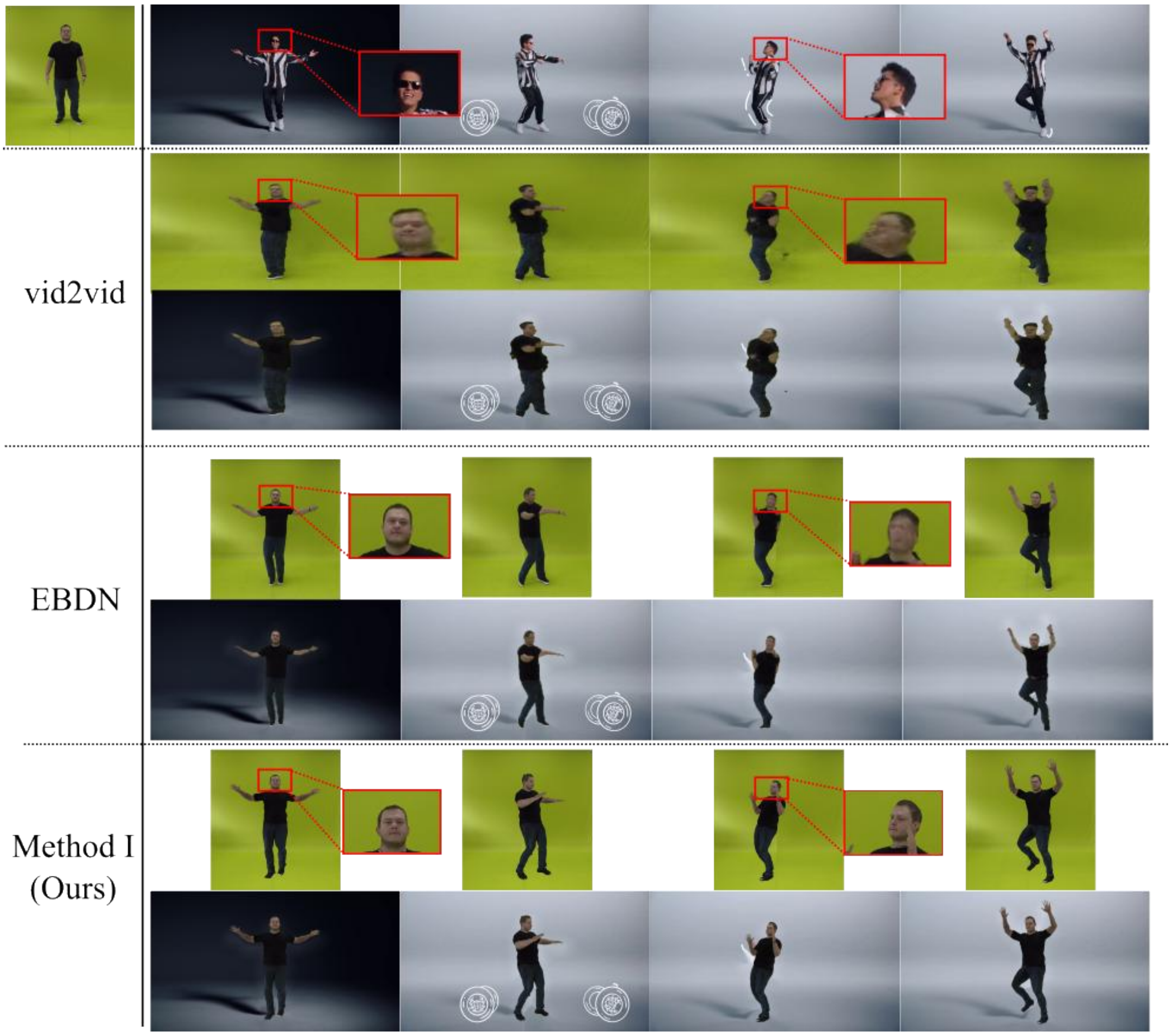

Figure 5.4: Method I: Qualitative evaluation to bruno-mars sequence. *First row:* original video and target actor S5; *Second row:* Result of vid2vid and their compositing with the target background; *Third row:* Results of EBDN and their compositing with the target background; Fourth row: Our results for both backgrounds.

reconstructed poorly the target's face. These strategies also incur a loss of the relative position in the original image; thus, whenever the actor presents a large translation, he/she can stay out of the crop area. Figure 5.5 depicts the results of “tom-cruise” sequence. In this sequence, the source actor has a large translation in all directions; thus, we include a margin in the image to allow EBDN and vid2vid to process it. Nevertheless, the difference in the actor's scale results in low quality for vid2vid and EBDN, which suggests their lack of generalization of these methods to different poses, and changes in camera viewpoint and intrinsic parameters. On its turn, our method did not suffer from problems caused by the camera and image resolution, and provided the best scores and visual results.

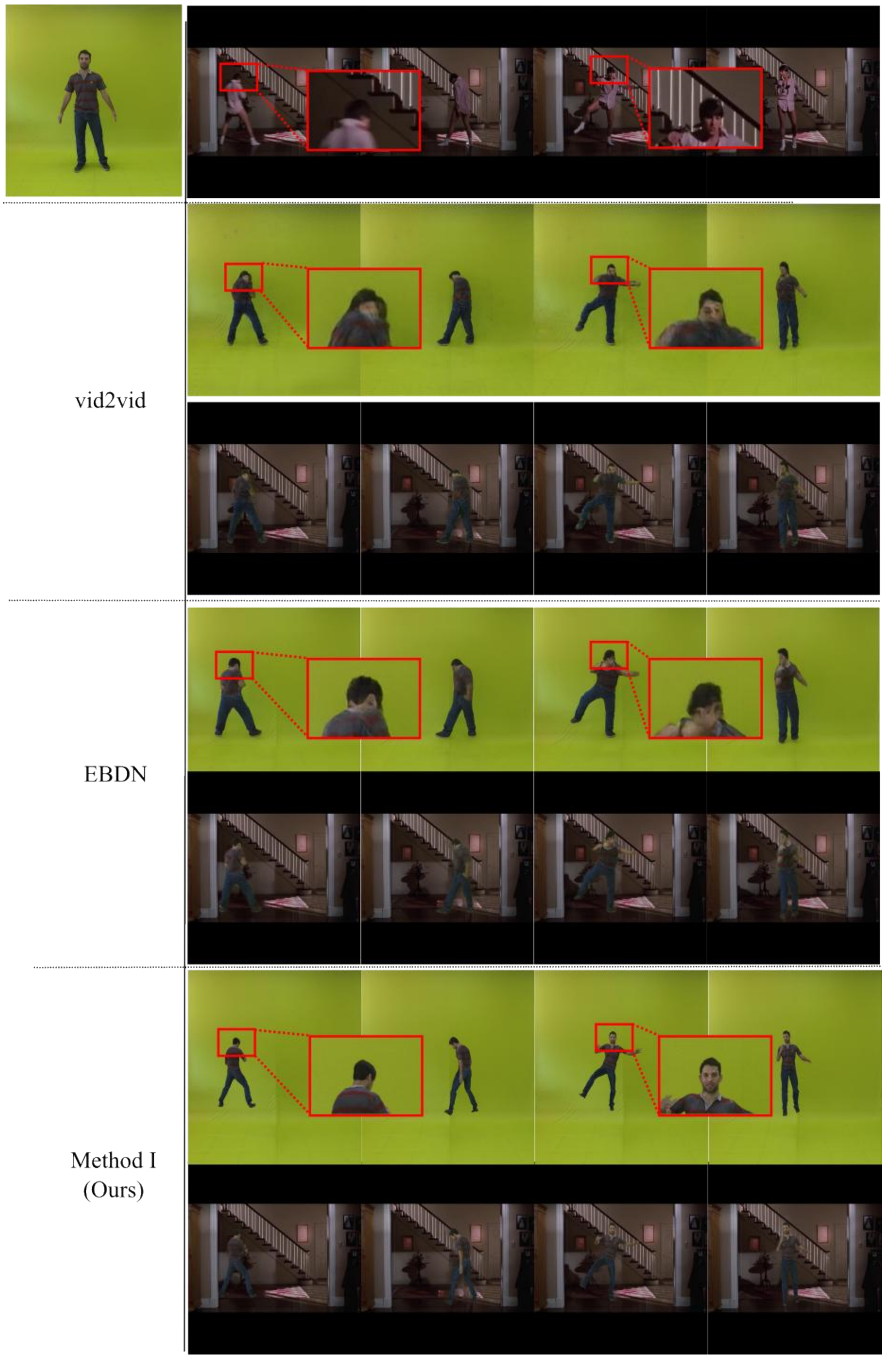

Figure 5.5: Method I: Qualitative evaluation to tom-cruise sequence. *First row:* original video and target actor S2; *Second row:* Result of vid2vid and their compositing with the target background; *Third row:* Result of EBDN and their compositing with the target background; *Fourth row:* Our results for both backgrounds.

### 5.2.6 Results in the iPER Dataset

Aside from our dataset, we also evaluated our approach using the iPER dataset [Liu et al., 2019]. Since the iPER dataset does not provide paired motions as the proposed dataset, we evaluated the Method I according to Liu et al. [2019]'s protocol, where SSIM and LPIPS are used to measure self-imitation transferring. We also include FVD in the evaluation, which was designed to capture the temporal coherence among videos and frame quality.

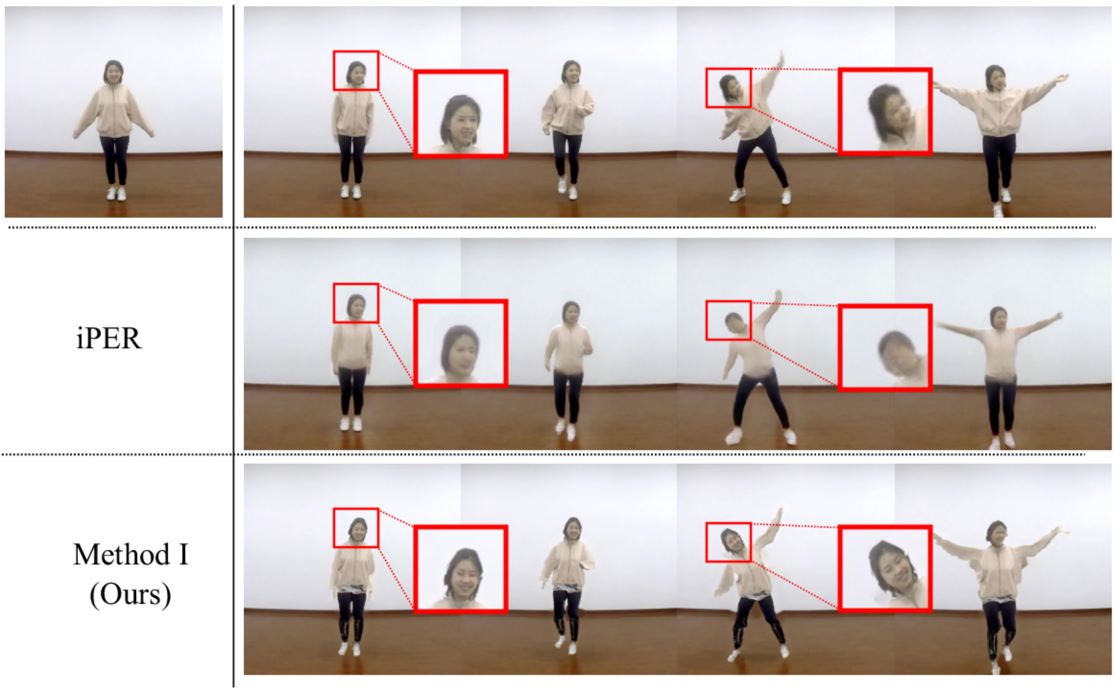

Figure 5.6: Method I: Qualitative evaluation on a sequence from the iPER dataset. *First row:* target actor and original video; *Second row:* Results of iPER network; *Third row:* Our results.

Table 5.7 shows the transferring results in the iPER dataset. It can be seen that Method I outperforms iPER in terms of SSIM and LPIPS metrics. This result also concurs with the visual results shown in Figure 5.6. We can also note that iPER achieved the best FVD value, since they are tested now in the same context where it was trained. However, it still performs poorly when the person is not standing up straight, as indicated in the facial zoom (red box in Figure 5.6). On the other hand, Method I retains the same quality for most poses.

### 5.2.7 Discussion

In the experiments, we observed that image-based rendering and 3D reasoning still have several advantages regarding end-to-end learning human view synthesis, particularly in terms of control and generalizing to furthest views. The designed first method enables the replacement of the background, which is crucial for many applications. Moreover,

Table 5.7: **Method I: Comparison with iPER in their proposed dataset**. Our approach is able to provide the best values in terms of SSIM and LPIPS, while performing worse in terms of FVD (best in bold).

| Metric | iPER | Method I (Ours) |
|---|---|---|
| SSIM↑ | 0.8410 | **0.8936** |
| LPIPS↓ | 0.0848 | **0.0722** |
| FVD↓ | **955** | 1269 |

it also allows to create a deformed model from images that can be included in virtual scenes using existing rendering frameworks, such as Blender. Figure 5.7 illustrates an application of this capability.

Regarding the method's processing time, our current implementation of the Method I is a modular Python code, but without any optimization, *i.e.*, the code was not parallelized either and was not adapted to run fully on a GPU. However, we highlight the available room for speeding up the processing time with a parallel implementation of different parts of the approach as, for instance, in the deformation model steps, which currently require 30 seconds per frame. They could be easily adapted to be executed in parallel with multiprocessing.

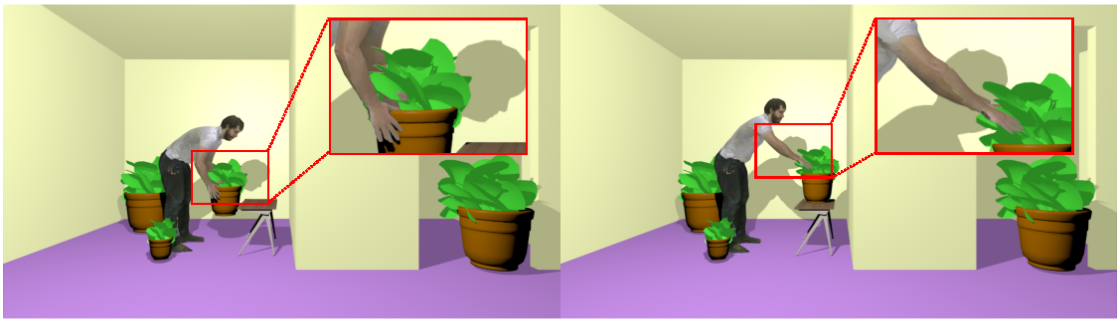

Figure 5.7: Method I: Model-to-virtual. Our method is able to provide a deformed model from images that can be included in virtual scenes using existing graphic rendering tools, such as Blender. The partial occlusions between the scene and model are handled in a natural way (red squares).

## 5.3 Method II: 3D Differentiable Human Rendering

### 5.3.1 Implementation details

We use PyTorch3D [Ravi et al., 2020] implementation of differentiable rendering and GCN operators. We trained our body mesh refinement network for 20 epochs with batch size equal to 4. For the optimizer, we used AdamW [Loshchilov and Hutter, 2019] with parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$, weight decay $= 1 \times 10^{-2}$, and learning rate of $1 \times 10^{-4}$ with a linear decay routine to zero starting from the middle of the training. We empirically set $\lambda_1$ and $\lambda_2$ to 1.0 and 0.5, respectively. In the Vertex refinement clamping component, we defined the set of thresholds as follows: $K \in \{$face $= 0.0$; footprints $= 0.0$; hands $= 0.0$; head $= 0.04$; torso $= 0.06$; arms $= 0.02$; forearms $= 0.04$; thighs $= 0.04$; calves $= 0.03$; feet $= 0.02\}$ meters.

Due to remarkable performance of pix2pix [Isola et al., 2017] in synthesizing photo-realistic images, we build our Texture Network upon its architecture. The optimizers for the texture models were configured as the same as the Mesh Network, except for the learning rates. The learning rate for the whole body and face discriminators, the global texture and refinement texture generators were set as $2 \times 10^{-5}$, $2 \times 10^{-5}$, $2 \times 10^{-3}$, and $2 \times 10^{-4}$, respectively. The parameters of the texture reconstruction was set to $\alpha_1 = 100$ and the regularization as $\alpha_2 = 100$. We observed that smaller values led to inconsistency in the final texture. For the training regime, we used 40 epochs with batch size 8. The global texture model was trained separately from the other models for 2,000 steps, then we freeze the model, the texture refinement generator and the discriminators were trained.

### 5.3.2 Processing Time

All the training and inference were performed in a single Titan XP (12 GB), where the GCN mesh model and the human texture networks took around 6 and 20 hours per actor, respectively. The inference time takes 92 ms per frame (90 ms in the GCN model deformation and 2 ms in the texture networks).

### 5.3.3 Training Setup

For the training of both texture models and the mesh human deformation network we considered four-minute videos provided by our dataset, where the actors perform random movements, allowing the model to get different views from the person in the scene. We use the SMPL model parameters calculated by our Motion Estimation method (3.2.1) and the silhouette image segmented by MODNet [Ke et al., 2020] for each frame of the video.

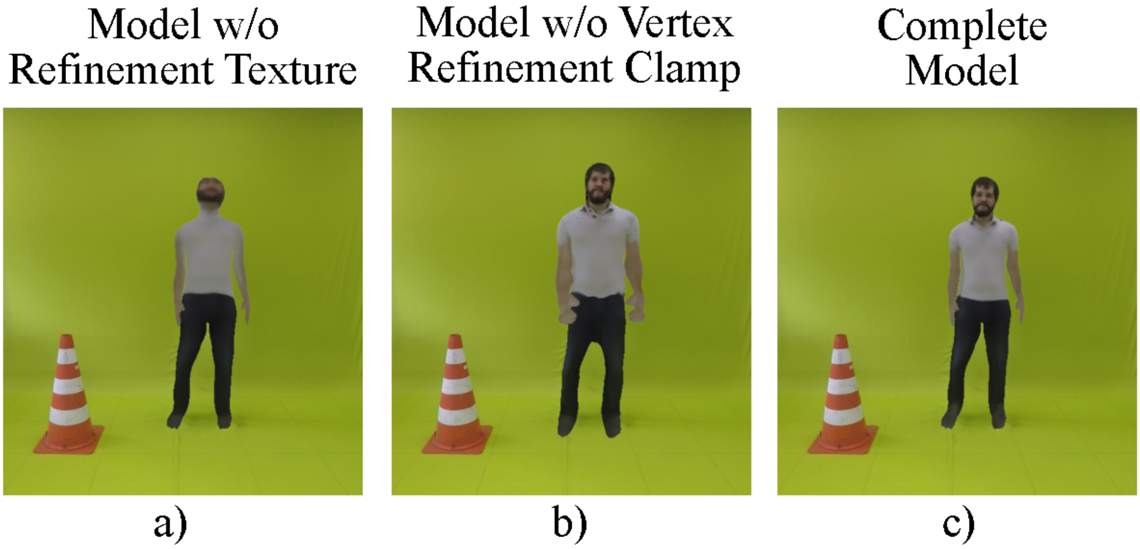
Model w/o Refinement Texture
Model w/o Vertex Refinement Clamp
Complete Model
a)
b)
c)

Figure 5.8: Method II: Ablation study. a) Results of the texture training without the refinement stage; b) Model without the Vertex Refinement Clamp layer. We observe an excessive growth of the mesh without update thresholds. The texture produced lacks details and even could not preserve the actor's face; c) shows the results for our complete model.

Table 5.8: **Method II: Ablation study**. SSIM, LPIPS, MSE, and FVD comparison by motion types. Best in bold.

| **Method** | **Metrics** | | | |
|---|---|---|---|---|
| | SSIM[1] | LPIPS[2] | MSE[2] | FVD[2] |
| Texture Refinement Removal | **0.869** | 0.136 | 262.39 | 795.15 |
| Vertex Refinement Clamping | 0.866 | 0.142 | 288.04 | 829.60 |
| Complete Model | 0.868 | **0.134** | **259.79** | **769.54** |

[1] *Higher is better* [2] *Lower is better*

### 5.3.4 Ablation Study

We evaluate the contributions of different parts of the method to the overall view synthesis performance. We investigated the benefits from the Vertex refinement clamping component in the Mesh Refinement Network (MRN) and the use of adversarial training in the texture generation. For the first experiment, we removed the vertex refinement thresholds, letting the mesh grow loosely. All other steps of texture training were maintained. Table 5.8 shows that the performance dropped drastically when compared to our original model. A qualitative analysis of the results in Figure 5.8-a demonstrates that removing the Vertex refinement clamping component led to strong wrong deformations in the hands and feet, *i.e.*, the regions with higher pose estimation errors.

In the adversarial training analysis, we maintained the original Mesh Refinement Network and removed the Texture Refinement Network and its discriminators, training only the Global Texture Network using Equation 3.11. Figure 5.8-b shows the texture quality of the models trained with and without the adversarial regime. After removing the GAN the model could not generate textures with fine details, producing blurry results. This result is also reported in the metrics of Table 5.8, where we show the average values calculated from all motion sequences in the test data in which the model without GAN is outperformed in all results besides SSIM. This result is coherent, since SSIM is based on low-level image features, and blurred textures can lead to higher SSIM.

### 5.3.5 Quantitative Comparison with State of The Art

We performed the neural rendering for actors with different body shapes, gender, clothing styles, and sizes for all considered video sequences. The video sequences used in the actor animation contained motions with different levels of difficulty, which aims to test the generalization capabilities of the methods in unseen data. Table 5.9 shows the performance for each method considering all motion and actors types in the dataset. We can see that our Method II achieves superior peformance as compared to the methods in most of the motion sequences and actor types considering the SSIM, LPIPS, MSE, and FVD metrics. We argue that these results indicate that our second method is capable of deforming the mesh according to the shape of the given actors and then, rendering a texture optimized to fit the morphology of the person in the scene. Furthermore, the training methodology, that considers multiple views from the actor and the shape parameters, allows the generation of consistent rendering with less deformations when the character is performing challenging movements, such as bending or rotating.

### 5.3.6 Qualitative Visual Analysis

The visual inspection of synthesized actors also concur with the quantitative analysis. In Figure 5.9, we provide the best frames for each movement using four actors in the dataset. Our Method I and our Method II are the only models capable of keeping the body scale of the authors along all scenes, while the other methods failed, in particular in the movements *shake hands* and *walk*. Besides generating coherent poses, our second method also generated more realistic textures in comparison to the other methods. Comparing the results of the movements *jump* and *spinning*, one can visualize some details as the shadow of the shirt sleeve of the actor and the shirt collar, respectively.

Table 5.9: **Method II: Comparison with state of the art**. SSIM, LPIPS, MSE, and FVD comparison by motion types and pair of actors from our dataset (best in bold, second-best in italic).

| Metric | Method | Motion type | | | | | | | | | Pair of actors | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | jump | walk | spinning | shake hands | cone | fusion dance | pull down | box | Avg. | S0-S1 | S2-S3 | S4-S5 | S6-S7 | Avg. |
| SSIM↑ | V-Unet | 0.870 | 0.871 | 0.843 | 0.847 | 0.862 | 0.797 | 0.847 | 0.857 | 0.849 | 0.855 | 0.886 | 0.830 | 0.826 | 0.849 |
| | Vid2Vid | 0.880 | 0.884 | 0.856 | 0.858 | 0.878 | 0.821 | 0.859 | 0.866 | 0.862 | 0.868 | 0.901 | 0.848 | 0.835 | 0.862 |
| | EBDN | 0.878 | 0.880 | 0.855 | 0.859 | 0.878 | 0.820 | 0.857 | 0.858 | 0.861 | 0.867 | 0.898 | 0.844 | 0.834 | 0.861 |
| | iPER | 0.877 | 0.880 | 0.852 | 0.859 | 0.877 | 0.816 | 0.855 | 0.856 | 0.859 | 0.867 | 0.896 | 0.842 | 0.831 | 0.859 |
| | Method I (Ours) | 0.881 | 0.885 | 0.855 | 0.860 | 0.879 | 0.820 | 0.861 | 0.869 | *0.864* | 0.872 | 0.902 | 0.846 | 0.834 | *0.864* |
| | Method II (Ours) | 0.884 | 0.890 | 0.860 | 0.865 | 0.885 | 0.824 | 0.866 | 0.873 | **0.868** | 0.876 | 0.908 | 0.852 | 0.838 | **0.868** |
| LPIPS↓ | V-Unet | 0.147 | 0.132 | 0.157 | 0.161 | 0.174 | 0.243 | 0.166 | 0.158 | 0.167 | 0.184 | 0.160 | 0.166 | 0.158 | 0.167 |
| | Vid2Vid | 0.131 | 0.105 | 0.126 | 0.136 | 0.133 | 0.203 | 0.142 | 0.133 | 0.138 | 0.148 | 0.131 | 0.129 | 0.147 | 0.138 |
| | EBDN | 0.141 | 0.122 | 0.139 | 0.138 | 0.143 | 0.215 | 0.151 | 0.170 | 0.153 | 0.159 | 0.145 | 0.147 | 0.159 | 0.153 |
| | iPER | 0.151 | 0.134 | 0.151 | 0.151 | 0.155 | 0.239 | 0.168 | 0.184 | 0.167 | 0.161 | 0.165 | 0.170 | 0.171 | 0.167 |
| | Method I (Ours) | 0.125 | 0.099 | 0.130 | 0.131 | 0.128 | 0.206 | 0.131 | 0.127 | *0.135* | 0.133 | 0.129 | 0.133 | 0.143 | *0.135* |
| | Method II (Ours) | 0.127 | 0.097 | 0.130 | 0.130 | 0.124 | 0.206 | 0.132 | 0.127 | **0.134** | 0.136 | 0.124 | 0.134 | 0.143 | **0.134** |
| MSE↓ | V-Unet | 295.04 | 269.59 | 354.33 | 377.02 | 328.68 | 559.75 | 417.00 | 346.13 | 368.44 | 381.77 | 344.71 | 362.63 | 384.66 | 368.44 |
| | Vid2Vid | 257.33 | 206.42 | 286.18 | 332.09 | 253.04 | 452.77 | 349.28 | 288.54 | 303.32 | 312.40 | 274.80 | 269.81 | 356.26 | 303.32 |
| | EBDN | 306.92 | 269.58 | 312.69 | 312.12 | 266.17 | 463.79 | 384.23 | 361.57 | 334.63 | 324.40 | 314.10 | 331.83 | 368.19 | 334.63 |
| | iPER | 313.43 | 275.22 | 344.94 | 314.92 | 267.39 | 504.00 | 404.79 | 377.16 | 350.23 | 277.98 | 328.07 | 358.30 | 436.57 | 350.23 |
| | Method I (Ours) | 237.16 | 178.57 | 286.86 | 270.25 | 237.86 | 434.64 | 301.66 | 245.86 | *274.11* | 243.95 | 260.14 | 294.88 | 297.46 | *274.11* |
| | Method II (Ours) | 231.20 | 153.23 | 278.75 | 254.38 | 218.76 | 418.58 | 286.02 | 237.42 | **259.79** | 247.10 | 236.49 | 276.17 | 279.42 | **259.79** |
| FVD↓ | V-Unet | 1,491.63 | 845.44 | 1,721.81 | 1,257.20 | 1,415.24 | 1,712.93 | 2,437.98 | 1,816.94 | 1,587.39 | 2,239.14 | 1,352.10 | 1,856.78 | 1,108.34 | 1,639.09 |
| | Vid2Vid | 879.94 | 266.6 | 1,085.49 | 396.31 | 790.79 | 997.42 | 997.96 | 1,069.85 | 810.48 | 778.53 | 719.80 | 762.46 | 574.08 | *708.72* |
| | EBDN | 887.56 | 273.00 | 918.94 | 423.08 | 725.49 | 952.22 | 1,113.46 | 853.26 | *768.37* | 791.98 | 751.45 | 560.27 | 826.71 | 732.60 |
| | iPER | 1,770.31 | 656.07 | 1,531.64 | 1,266.14 | 1,051.42 | 1,322.72 | 1,440.94 | 1,719.55 | 1,344.84 | 1,270.41 | 1,092.82 | 1,395.81 | 1,214.64 | 1,243.42 |
| | Method I (Ours) | 1,119.50 | 330.91 | 674.99 | 478.93 | 767.68 | 791.01 | 988.35 | 760.33 | **738.96** | 715.00 | 653.30 | 720.49 | 515.61 | **651.10** |
| | Method II (Ours) | 1,114.43 | 233.81 | 1019.83 | 542.24 | 614.88 | 746.22 | 1010.24 | 874.69 | 769,54 | 881.49 | 697.68 | 718.21 | 551.06 | 712.11 |

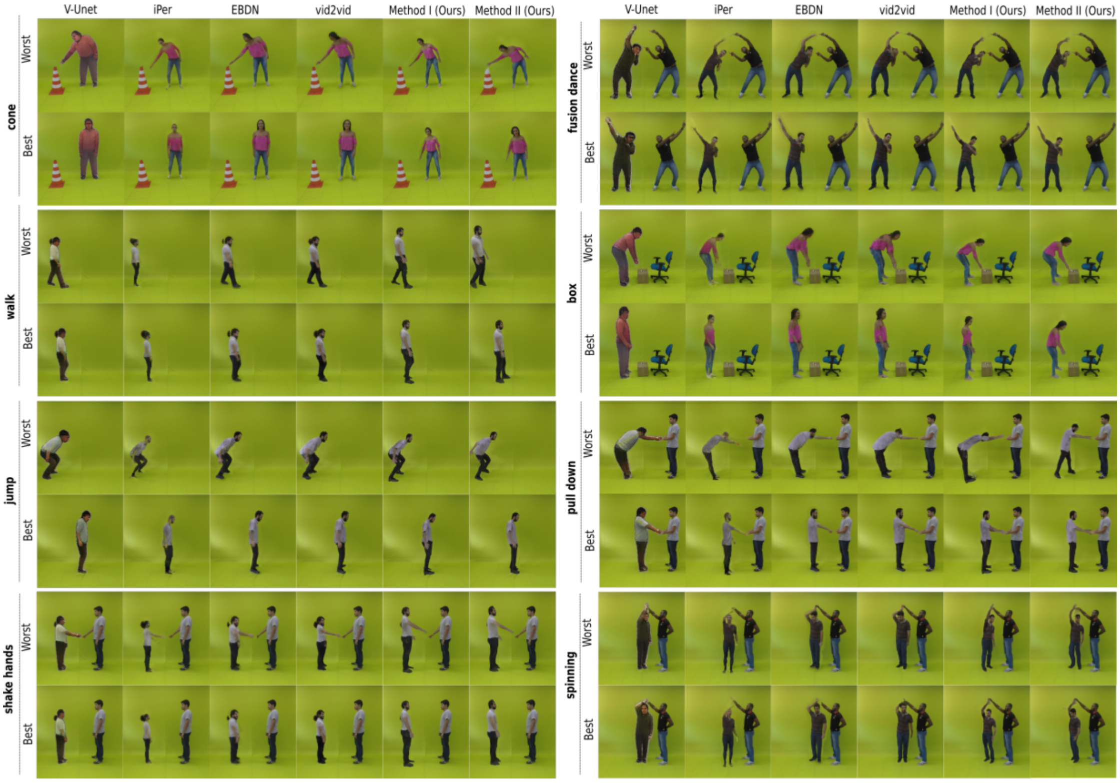

Figure 5.9: Method II: Qualitative comparison. Transferring results considering the cases where the person is not standing parallel to the image plane or has the arms in front of the face. In each sequence: the first row shows the worst generated frame for each method and the second row presents the best generated frame for each method.

The Figure 5.10 illustrates a task of retargeting in two different scenarios. These results demonstrate the capability of generating detailed face and body texture, producing a good fit of the actors in the different scenes.

### 5.3.7 Discussion

The experiments show that the proposed method superior quality compared to recent neural rendering techniques, besides having several advantages in terms of control and ability to generalize to furthest views. Yet we observed none of the existing methods obtain artifact-free results, which suggests the problem of synthesizing realistic views of people in general contexts is still a challenging problem. On the other hand, the results indicate that neural networks and differentiable rendering approaches have the potential to push forward the area.

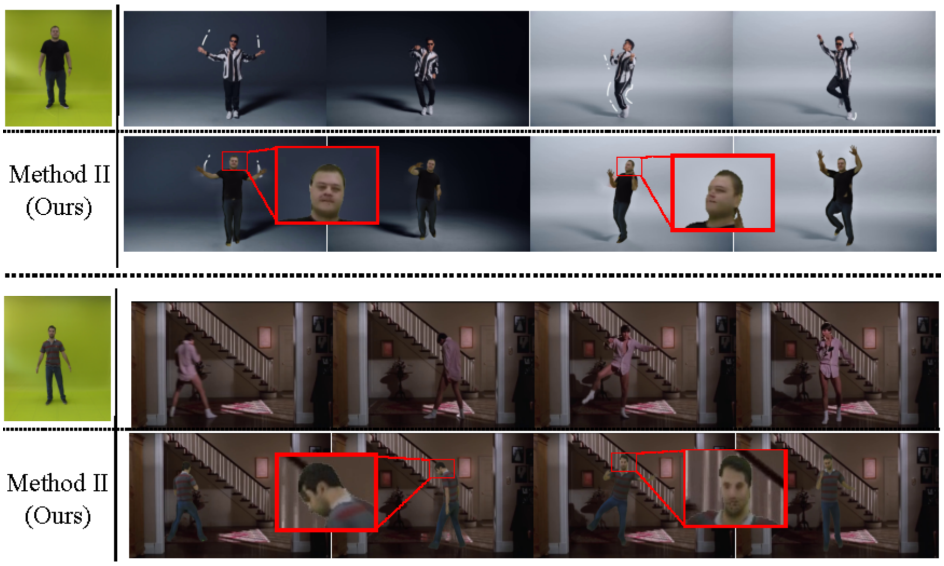

Figure 5.10: Method II: Human retargeting example. The first line of each scene illustrates the real movement. On the second line is the retargeting using our proposed method. The red squares highlight our face generation quality.

## 5.4 Conclusions and Closing Remarks

This chapter presented several experiments that we performed to show the behavior of our two methods. We performed a comparative analysis in terms of motion, shape, and appearance quality against the literature's recent 2D neural rendering methods. In these experiments, the *Image-Based Rendering* method and the *3D Differentiable Human Rendering* method outperformed the other approaches, including data-driven methods like EBDN and vid2vid.

Our strategy that simultaneously considers body shape, motion retargeting constraints, and visual appearance quality results in methods less sensitive to the camera and pose conditions. As shown in the experiments, our approaches are both stable and shape-aware. In other words, they do not suffer from quality instability when applied in contexts slightly different from the original ones (a small difference in camera position, uncommon motions, pose translation, *etc.*), and they can handle different morphologies in the retargeting. Moreover, the proposed methodology has several advantages in terms of control and adaptability to new contexts. For example, our methods can easily replace the background and incorporate a tracking algorithm to work appropriately in scenes with two or more people, which is impossible in previous work since they embed the character into the same learned background.

As shown in the experiments, none of the existing methods obtain artifact-free results, which suggests the problem of synthesizing realistic views of people in general contexts is still a challenging problem. Moreover, aspects such as illumination and temporal harmonization still are neglected by all methods. They are essential aspects of achieving realism.

We remark that it would be pertinent further to analyze the local quality of the retargeting methods. The metrics adopted in this dissertation and the literature compute the global quality of a synthesized frame. However, the background corresponds to most of the image region, and it is known that facial abnormalities draw the attention of humans, which is not considered in the metrics.

# Chapter 6

# Conclusions

This dissertation proposes a general methodology of transferring human motion and appearance from video to video preserving motion features, body shape, and visual quality. We designed two novel methods using our methodology presented in Chapter 3 and we demonstrated that this methodology is adequate to be used as a design guide in the creation of new methods to transfer human motion and appearance from video to video preserving motion features, body shape, and visual quality. From a theorical standpoint, our work exploits motion constraints, body shape, and a 3D representation of people to synthesizing more plausible videos and allows us to tackle subjects with different limb proportions and body shape. Thereby, this work offers three main contributions to the state of the art:

i. A unified methodology carefully designed to transfer motion and appearance from video to video that preserves the main features of the human movement and retains the visual appearance of the target character;

ii. A retargeting technique taking into account physical constraints of the motion in 3D and the image domain; and new semantic-guided image-based rendering approach that copies local patterns from input images to the correct position in the generated images, which defines a more stable method and overcome the lack of details;

iii. A novel data-driven formulation for transferring appearance and reenact human actors that produces a fully 3D controllable human model, *i.e.*, the user can control the human pose and rendering parameters.

We performed experiments on different publicly available videos and on a dedicated collected dataset containing several types of motions, constraints, and actors with

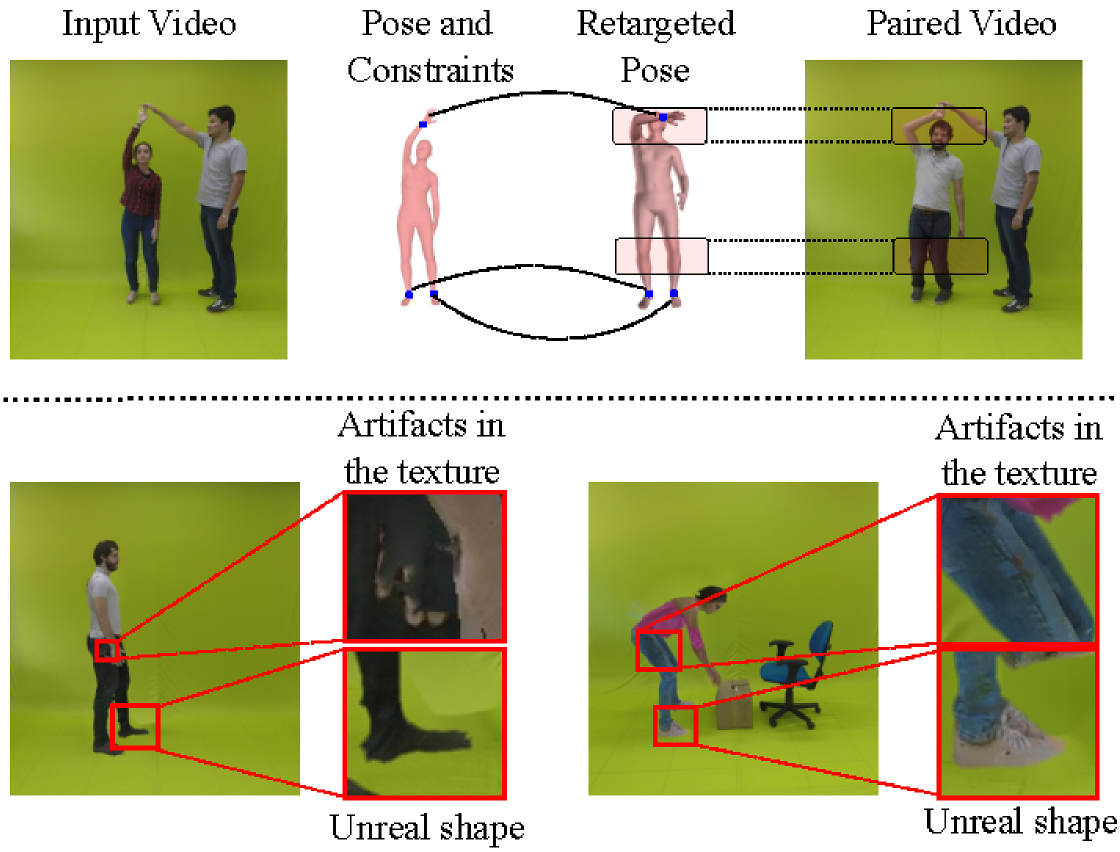

Figure 6.1: Limitations. *Top:* Retargeting resulting in undesired motion where the actor positions his hand down and curves his back instead of bending his knees. *Bottom:* Typical failure cases in the avatar: artifacts in the texture and body parts with unreal shapes.

different body shapes. Our approaches achieved better results than learning methodologies like the EBDN and vid2vid in most scenarios for appearance metrics (SSIM, LPIPS, MSE, FVD). Our results also indicate that retargeting strategies based on image-to-image learning are still challenged to retarget motions while keeping the desired movement constraints, shape, and appearance simultaneously. Furthermore, we show in our second method that it is possible to build hybrid strategy that produces a fully 3D representation of the person. By taking advantages of both differentiable rendering and the 3D parametric model, this approaches allow controlling the human pose and rendering parameters.

## 6.1 Future Work

Although achieving the best results in these more generic test conditions, the proposed approaches also suffer from certain limitations. Ideally, the constrained optimization problem would maintain the main features of the source motion. However, there is no single solution to a set of constraints, which can result in undesired motions, as shown in Figure 6.1, where the actor positions his hand down and curve his back instead of bending his knees. One possibility to increase the robustness of this method would be using simultaneously kinematic and dynamic constraints such as self-collisions and balance. Another interesting topic is how to reuse other existing sources of motion. If we want a human character to imitate a dog's walking, there is no one-to-one correspondence between the bones of their skeletons. Capture systems may define different skeletons even if we are dealing with the same type of characters. Thus, transferring stylistic motion features between characters with different skeletal topologies might be an exciting topic.

The textured avatars built by our first method may exhibit artifacts in the presence of part segmentation estimation errors, which can also lead to wrong deformations, and errors in the deformation result in body parts with unreal shapes. Segmentation errors are also depicted in Figure 6.1. The results of our second method indicate that neural networks and differentiable rendering approaches can push forward the area and reduce the artifacts. However, it is challenging to train GANs to produce mesh and texture due to training instability and optimization issues. These challenges call for more robust loss functions and stable training procedures. In the future, end-to-end training with more compact, simple, and memory-efficient network architectures is a potential next step in this direction. Also, the results shown in the experiments chapter have demonstrated the importance of using an appropriate strategy to combine human motion and appearance transferring. We believe it is important and necessary to proceed with a theoretical investigation about the limits and best ways to perform such combinations.

Another important direction in the future is to address the composition step. We would like to try a strategy to estimate the light of the target scene and the material properties of the model. Thus, removing the inconsistency between the foreground and background would be possible by applying the correct shading and shadows to the scene, which is impossible to our methods.

Although differentiable rendering is a novel field, it is rapidly maturing, aided by the continuous development of new tools to simplify its usage. This will enable more researchers to develop hybrid approaches that combine the best of model-based and learning-based approaches. Differentiable rendering of videos is also an exciting research direction to be explored, in order to train an end-to-end pipeline that combines video data with motion constraints.

# Bibliography

Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., and Chen, B. (2020). Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)*, 39(4):62.

Aberman, K., Shi, M., Liao, J., Lischinski, D., Chen, B., and Cohen-Or, D. (2018). Deep video-based performance cloning. *CoRR*.

Aberman, K., Wu, R., Lischinski, D., Chen, B., and Cohen-Or, D. (2019). Learning character-agnostic motion for motion retargeting in 2d. *ACM Transactions on Graphics (TOG)*, 38(4):75.

Akhter, I. and Black, M. J. (2015). Pose-conditioned joint angle limits for 3D human pose reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015)*, pages 1446--1455.

Alldieck, T., Magnor, M., Xu, W., Theobalt, C., and Pons-Moll, G. (2018). Video based reconstruction of 3d people models. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). Scape: Shape completion and animation of people. *ACM Trans. Graph.*

Arikan, O. and Forsyth, D. A. (2002). Interactive motion generation from examples. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 483--490, New York, NY, USA. ACM.

Balakrishnan, G., Zhao, A., Dalca, A. V., Durand, F., and Guttag, J. V. (2018). Synthesizing images of humans in unseen poses. In *CVPR*.

Barron-Romero, C. and Kakadiaris, I. (2001). Estimating anthropometry and pose from a single uncalibrated image. *Computer Vision and Image Understanding*, 81:269–284.

Bau, D., Zhu, J.-Y., Wulff, J., Peebles, W., Strobelt, H., Zhou, B., and Torralba, A. (2019). Seeing what a gan cannot generate. In *ICCV*.

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

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*.

Chan, C., Ginosar, S., Zhou, T., and Efros, A. (2019). Everybody dance now. In *ICCV*.

Chen, L., Day, T. W., Tang, W., and John, N. W. (2017). Recent developments and future challenges in medical mixed reality. In Broll, W., Regenbrecht, H., and II, J. E. S., editors, *2017 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2017, Nantes, France, October 9-13, 2017*, pages 123--135. IEEE Computer Society.

Choi, K.-J. and Ko, H.-S. (2000). On-line motion retargeting. *Journal of Visualization and Computer Animation*.

Criminisi, A., Perez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based image inpainting. *IEEE TIP*.

De Boor, C., De Boor, C., Mathématicien, E.-U., De Boor, C., and De Boor, C. (1978). *A practical guide to splines*, volume 27.

Dosovitskiy, A., Springenberg, J. T., and Brox, T. (2015). Learning to generate chairs with convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1538–1546. ISSN 1063-6919.

Esser, P., Sutter, E., and Ommer, B. (2018). A variational u-net for conditional appearance and shape generation. In *CVPR*.

Ferreira, J. P., Coutinho, T. M., Gomes, T. L., Neto, J. F., Azevedo, R., Martins, R., and Nascimento, E. R. (2021). Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. *Computers & Graphics*, 94:11 – 21.

Gallala, A., Hichri, B., and Plapper, P. (2019). Survey: The evolution of the usage of augmented reality in industry 4.0. *IOP Conference Series: Materials Science and Engineering*, 521:012017.

Gkioxari, G., Malik, J., and Johnson, J. (2019). Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Gleicher, M. (1998). Retargetting motion to new characters. In *SIGGRAPH*.

Gleicher, M. (2001). Motion path editing. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, pages 195--202, New York, NY, USA. ACM.

Gong, K., Liang, X., Li, Y., Chen, Y., Yang, M., and Lin, L. (2018). Instance-level human parsing via part grouping network. In *ECCV*.

Gong, W., Zhang, X., GonzÃ lez, J., Sobral, A., Bouwmans, T., Tu, C., and ZAHZAH, E.-h. (2016). Human pose estimation from monocular images: A comprehensive survey. *Sensors*, 16:1966.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672--2680.

Hasler, N., Ackermann, H., Rosenhahn, B., Thormahlen, T., and Seidel, H.-P. (2010). Multilinear pose and body shape estimation of dressed subjects from image sets. pages 1823 – 1830.

Hassan, M., Choutas, V., Tzionas, D., and Black, M. J. (2019). Resolving 3D human pose ambiguities with 3D scene constraints. In *ICCV*.

Huang, Z., Xu, Y., Lassner, C., Li, H., and Tung, T. (2020). ARCH: animatable reconstruction of clothed humans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

InterDigital (2020). The sustainable future of video entertainment.

Ionescu, C., Carreira, J., and Sminchisescu, C. (2014a). Iterated second-order label sensitive pooling for 3d human pose estimation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1661--1668, Washington, DC, USA. IEEE Computer Society.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014b). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *CVPR*.

Kanamori, Y. and Endo, Y. (2018). Relighting humans: Occlusion-aware inverse rendering for full-body human images. *ACM Trans. Graph.*, 37(6):270:1--270:11. ISSN 0730-0301.

Kanazawa, A., Black, M. J., Jacobs, D. W., and Malik, J. (2018). End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Regognition (CVPR)*.

Kang, S. B. and Shum, H.-Y. (2000). A review of image-based rendering techniques.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*.

Kato, H., Beker, D., Morariu, M., Ando, T., Matsuoka, T., Kehl, W., and Gaidon, A. (2020). Differentiable rendering: A survey. *arXiv preprint*.

Kato, H., Ushiku, Y., and Harada, T. (2018). Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ke, Z., Li, K., Zhou, Y., Wu, Q., Mao, X., Yan, Q., and Lau, R. W. (2020). Is a green screen really necessary for real-time portrait matting? *ArXiv*, abs/2011.11961.

Kolotouros, N., Pavlakos, G., Black, M. J., and Daniilidis, K. (2019). Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*.

Kostrikov, I. and Gall, J. (2014). Depth sweep regression forests for estimating 3d human pose from images. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

Lassner, C., Pons-Moll, G., and Gehler, P. V. (2017a). A generative model for people in clothing. In *Proceedings of the IEEE International Conference on Computer Vision*.

Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., and Gehler, P. V. (2017b). Unite the people: Closing the loop between 3d and 2d human representations. In *CVPR*.

Lazova, V., Insafutdinov, E., and Pons-Moll, G. (2019). 360-degree textures of people in clothing from a single image. In *2019 International Conference on 3D Vision, 3DV 2019, Québec City, QC, Canada, September 16-19, 2019*, pages 643--653. IEEE.

Lee, H.-Y., Yang, X., Liu, M.-Y., Wang, T.-C., Lu, Y.-D., Yang, M.-H., and Kautz, J. (2019). Dancing to music. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Lee, J. and Shin, S. Y. (1999). A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 39--48, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Levi, Z. and Gotsman, C. (2015). Smooth rotation enhanced as-rigid-as-possible mesh animation. *T-VCG.*

Li, S. and Chan, A. (2014). 3d human pose estimation from monocular images with deep convolutional neural network. volume 9004, pages 332–347.

Li, Y., Wang, T., and Shum, H.-Y. (2002). Motion texture: A two-level statistical model for character motion synthesis. volume 21, pages 465–472.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2014). Microsoft COCO: Common Objects in Context. *arXiv e-prints*, page arXiv:1405.0312.

Liu, L., Xu, W., Zollhoefer, M., Kim, H., Bernard, F., Habermann, M., Wang, W., and Theobalt, C. (2019a). Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics (TOG)*, 38(5):1--14.

Liu, S., Li, T., Chen, W., and Li, H. (2019b). Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV).*

Liu, W., Piao, Z., Jie, M., Luo, W., Ma, L., and Gao, S. (2019). Liquid warping GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV.*

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*

Loper, M. M. and Black, M. J. (2014). Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Ma, L., Jia, X., Sun, Q., Schiele, B., Tuytelaars, T., and Van Gool, L. (2017). Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 405--415.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. (2019). AMASS: Archive of motion capture as surface shapes. In *ICCV*.

Marra, F., Gragnaniello, D., Verdoliva, L., and Poggi, G. (2020). A full-image full-resolution end-to-end-trainable cnn framework for image forgery detection. *IEEE Access*.

Martinez, J., Hossain, R., Romero, J., and Little, J. J. (2017). A simple yet effective baseline for 3d human pose estimation. In *ICCV*.

Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., and Theobalt, C. (2017). Monocular 3d human pose estimation in the wild using improved cnn supervision. pages 506–516.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2020). Image segmentation using deep learning: A survey.

Mir, A., Alldieck, T., and Pons-Moll, G. (2020). Learning to transfer texture from clothing images to 3d humans. In *CVPR*. IEEE.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*, abs/1411.1784.

Natsume, R., Saito, S., Huang, Z., Chen, W., Ma, C., Li, H., and Morishima, S. (2019). Siclope: Silhouette-based clothed people. In *CVPR*.

Neverova, N., Güler, R. A., and Kokkinos, I. (2018). Dense pose transfer. In *ECCV*.

Niemeyer, M., Mescheder, L., Oechsle, M., and Geiger, A. (2020). Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Parameswaran, V. and Chellappa, R. (2004). View independent human body pose estimation from a single perspective image. volume 2, pages II–16.

Peng, X. B., Kanazawa, A., Malik, J., Abbeel, P., and Levine, S. (2018). Sfv: Reinforcement learning of physical skills from videos. *ACM Trans. Graph.*, 37(6).

Pérez, P., Gangnet, M., and Blake, A. (2003). Poisson image editing. *ACM Trans. Graph.*, 22(3):313--318. ISSN 0730-0301.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Ramakrishna, V., Kanade, T., and Sheikh, Y. (2012). Reconstructing 3d human pose from 2d image landmarks. volume 7575, pages 573–586.

Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.-Y., Johnson, J., and Gkioxari, G. (2020). Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*.

Ren, X., Li, H., Huang, Z., and Chen, Q. (2020). Self-supervised dance video synthesis conditioned on music. In *ACM MM*.

Riza Alp Güler, Natalia Neverova, I. K. (2018). Densepose: Dense human pose estimation in the wild. *arXiv*.

Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., and Li, H. (2019). PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *IEEE/CVF International Conference on Computer Vision*.

Saito, S., Simon, T., Saragih, J., and Joo, H. (2020). PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Sandvine (2019). Global internet phenomena.

Shum, H.-Y., Kang, S. B., and Chan, S.-C. (2003). Survey of image-based representations and compression techniques. *TCSVT*.

Shysheya, A., Zakharov, E., Aliev, K.-A., Bashirov, R., Burkov, E., Iskakov, K., Ivakhnenko, A., Malkov, Y., Pasechnik, I., Ulyanov, D., Vakhitov, A., and Lempitsky, V. (2019). Textured neural avatars. In *CVPR*.

Sidenbladh, H., De la Torre, F., and Black, M. J. (2000). A framework for modeling the appearance of 3d articulated figures. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 368–375. ISSN .

Sigal, L., Balan, A., and Black, M. J. (2007). Combined discriminative and generative articulated pose and non-rigid shape estimation. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1337--1344, USA. Curran Associates Inc.

Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*.

Sun, Y.-T., Fu, Q.-C., Jiang, Y.-R., Liu, Z., Lai, Y.-K., Fu, H., and Gao, L. (2020). Human motion transfer with 3d constraints and detail enhancement.

Tak, S. and Ko, H.-S. (2005). A physically-based motion retargeting filter. *ACM Trans. Graph.*, 24(1):98--117. ISSN 0730-0301.

Tatarchenko, M., Dosovitskiy, A., and Brox, T. (2015). Single-view to multi-view: Reconstructing unseen views with a convolutional network. *CoRR*, abs/1511.06702.

Tekin, B., Rozantsev, A., Lepetit, V., and Fua, P. (2015). Direct prediction of 3d body poses from motion compensated sequences. *CoRR*, abs/1511.06692.

Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J., Niebner, M., Pandey, R., Fanello, S., Wetzstein, G., Zhu, J.-Y., Theobalt, C., Agrawala, M., Shechtman, E., Goldman, D. B., and Zollhofer, M. (2020). State of the art on neural rendering. *Computer Graphics Forum*, 39(2):701–727.

Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., and Gelly, S. (2019). Towards accurate generative models of video: A new metric & challenges.

Villegas, R., Yang, J., Ceylan, D., and Lee, H. (2018). Neural kinematic networks for unsupervised motion retargetting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, C., Huang, H., Han, X., and Wang, J. (2019). Video inpainting by jointly learning temporal structure and spatial details. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence.*

Wang, J., Yan, S., Xiong, Y., and Lin, D. (2020a). Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision.*

Wang, S.-Y., Wang, O., Zhang, R., Owens, A., and Efros, A. A. (2020b). Cnn-generated images are surprisingly easy to spot... for now. In *CVPR.*

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. (2018). Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS).*

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 13(4):600--612.

Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *CVPR.*

Wu, W., Zhang, Y., Li, C., Qian, C., and Loy, C. C. (2018). Reenactgan: Learning to reenact faces via boundary transfer. In *ECCV.*

Yan, S., Li, Z., Xiong, Y., Yan, H., and Lin, D. (2019). Convolutional sequence generation for skeleton-based action synthesis. In *IEEE/CVF International Conference on Computer Vision.*

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI conference on artificial intelligence.*

Yang, J., Reed, S., Yang, M.-H., and Lee, H. (2015). Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1099--1107, Cambridge, MA, USA. MIT Press.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2018). Generative image inpainting with contextual attention. In *Computer Vision and Pattern Recognition (CVPR).*

Zhang, C. and Chen, T. (2004). A survey on image-based rendering-representation, sampling and compression. *Signal Processing: Image Communication.*

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018a). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

Zhang, S., Tong, H., Xu, J., and Maciejewski, R. (2018b). Graph convolutional networks: Algorithms, applications and open challenges. In Chen, X., Sen, A., Li, W. W., and Thai, M. T., editors, *Computational Data and Social Networks*, pages 79--91, Cham. Springer International Publishing.

Zhang, Y., Chen, W., Ling, H., Gao, J., Zhang, Y., Torralba, A., and Fidler, S. (2020). Image GANs meet differentiable rendering for inverse graphics and interpretable 3D neural rendering. *arXiv preprint arXiv:2010.09125*.

Zhao, L., Peng, X., Tian, Y., Kapadia, M., and Metaxas, D. N. (2019). Semantic graph convolutional networks for 3d human pose regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zhou, T., Tulsiani, S., Sun, W., Malik, J., and Efros, A. A. (2016a). View synthesis by appearance flow. *CoRR*, abs/1605.03557.

Zhou, X., Sun, X., Zhang, W., Liang, S., and Wei, Y. (2016b). Deep kinematic pose regression. In *ECCV Workshops (3)*, volume 9915 of *Lecture Notes in Computer Science*, pages 186--201.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.