REAL-TIME HUMAN PERFORMANCE CAPTURE AND SYNTHESIS

Dissertation zur Erlangung des Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

der Fakultät für Mathematik und Informatik
der Universität des Saarlandes

MARC HABERMANN

Saarbrücken, 2021

UNIVERSITÄT
DES
SAARLANDES

Date of Colloquium:
October 29, 2021

Dean of the Faculty:
Prof. Dr. Thomas Schuster

Chair of the Committee:
Prof. Dr. Jürgen Steimle

Reviewers:
Prof. Dr. Christian Theobalt
Prof. Dr. Hans-Peter Seidel
Prof. Dr. Adrian Hilton

Academic Assistant:
Dr. Lingjie Liu

Für
meine Eltern und Geschwister,
die es mir ermöglicht haben meine Träume zu verwirklichen
und
meine Frau,
die jeden Tag zum Traum werden lässt.

ABSTRACT

Most of the images one finds in the media, such as on the Internet or in textbooks and magazines, contain humans as the main point of attention. Thus, there is an inherent necessity for industry, society, and private persons to be able to thoroughly analyze and synthesize the human-related content in these images.

One aspect of this analysis and subject of this thesis is to infer the 3D pose and surface deformation, using only visual information, which is also known as *human performance capture*. Human performance capture enables the tracking of virtual characters from real-world observations, and this is key for visual effects, games, VR, and AR, to name just a few application areas. However, traditional capture methods usually rely on expensive multi-view (marker-based) systems that are prohibitively expensive for the vast majority of people, or they use depth sensors, which are still not as common as single color cameras. Recently, some approaches have attempted to solve the task by assuming only a single RGB image is given. Nonetheless, they can either not track the dense deforming geometry of the human, such as the clothing layers, or they are far from real time, which is indispensable for many applications. To overcome these shortcomings, this thesis proposes two monocular human performance capture methods, which for the first time allow the real-time capture of the dense deforming geometry as well as an unseen 3D accuracy for pose and surface deformations. At the technical core, this work introduces novel GPU-based and data-parallel optimization strategies in conjunction with other algorithmic design choices that are all geared towards real-time performance at high accuracy. Moreover, this thesis presents a new weakly supervised multi-view training strategy combined with a fully differentiable character representation that shows superior 3D accuracy.

However, there is more to human-related Computer Vision than only the analysis of people in images. It is equally important to synthesize new images of humans in unseen poses and also from camera viewpoints that have not been observed in the real world. Such tools are essential for the movie industry because they, for example, allow the synthesis of photo-realistic virtual worlds with real-looking humans or of contents that are too dangerous for actors to perform on set. But also video conferencing and telepresence applications can benefit from photo-real 3D characters, as they can enhance the immersive experience of these applications. Here, the traditional Computer Graphics pipeline for rendering photo-realistic images involves many tedious and time-consuming steps that require expert knowledge and are far from real time. Traditional rendering involves character rigging

and skinning, the modeling of the surface appearance properties, and physically based ray tracing. Recent learning-based methods attempt to simplify the traditional rendering pipeline and instead learn the rendering function from data resulting in methods that are easier accessible to non-experts. However, most of them model the synthesis task entirely in image space such that 3D consistency cannot be achieved, and/or they fail to model motion- and view-dependent appearance effects. To this end, this thesis presents a method and ongoing work on character synthesis, which allow the synthesis of controllable photo-real characters that achieve motion- and view-dependent appearance effects as well as 3D consistency and which run in real time. This is technically achieved by a novel coarse-to-fine geometric character representation for efficient synthesis, which can be solely supervised on multi-view imagery. Furthermore, this work shows how such a geometric representation can be combined with an implicit surface representation to boost synthesis and geometric quality.

ZUSAMMENFASSUNG

In den meisten Bildern in den heutigen Medien, wie dem Internet, Büchern und Magazinen, ist der Mensch das zentrale Objekt der Bildkomposition. Daher besteht eine inhärente Notwendigkeit für die Industrie, die Gesellschaft und auch für Privatpersonen, die auf den Mensch fokussierten Eigenschaften in den Bildern detailliert analysieren und auch synthetisieren zu können.

Ein Teilaspekt der Anaylse von menschlichen Bilddaten und damit Bestandteil der Thesis ist das Rekonstruieren der 3D-Skelett-Pose und der Oberflächendeformation des Menschen anhand von visuellen Informationen, was fachsprachlich auch als Human Performance Capture bezeichnet wird. Solche Rekonstruktionsverfahren ermöglichen das Tracking von virtuellen Charakteren anhand von Beobachtungen in der echten Welt, was unabdingbar ist für Applikationen im Bereich der visuellen Effekte, Virtual und Augmented Reality, um nur einige Applikationsfelder zu nennen. Nichtsdestotrotz basieren traditionelle Tracking-Methoden auf teuren (markerbasierten) Multi-Kamera Systemen, welche für die Mehrheit der Bevölkerung nicht erschwinglich sind oder auf Tiefenkameras, die noch immer nicht so gebräuchlich sind wie herkömmliche Farbkameras. In den letzten Jahren gab es daher erste Methoden, die versuchen, das Tracking-Problem nur mit Hilfe einer Farbkamera zu lösen. Allerdings können diese entweder die Kleidung der Person im Bild nicht tracken oder die Methoden benötigen zu viel Rechenzeit, als dass sie in realen Applikationen genutzt werden könnten. Um diese Probleme zu lösen, stellt die Thesis zwei monokulare Human Performance Capture Methoden vor, die zum ersten Mal eine Echtzeit-Rechenleistung erreichen sowie im Vergleich zu vorherigen Arbeiten die Genauigkeit von Pose und Oberfläche in 3D weiter verbessern. Der Kern der Methoden beinhaltet eine neuartige GPU-basierte und datenparallelisierte Optimierungsstrategie, die im Zusammenspiel mit anderen algorithmischen Designentscheidungen akkurate Ergebnisse erzeugt und dabei eine Echtzeit-Laufzeit ermöglicht. Daneben wird eine neue, differenzierbare und schwach beaufsichtigte, Multi-Kamera basierte Trainingsstrategie in Kombination mit einem komplett differenzierbaren Charaktermodell vorgestellt, welches ungesehene 3D Präzision erreicht.

Allerdings spielt nicht nur die Analyse von Menschen in Bildern in Computer Vision eine wichtige Rolle, sondern auch die Möglichkeit, neue Bilder von Personen in unterschiedlichen Posen und Kamera-Blickwinkeln synthetisch zu rendern, ohne dass solche Daten zuvor in der Realität aufgenommen wurden. Diese Methoden sind unabdingbar für die Filmindustrie, da sie es zum Beispiel ermöglichen, fotorealis-

tische virtuelle Welten mit real aussehenden Menschen zu erzeugen, sowie die Möglichkeit bieten, Szenen, die für den Schauspieler zu gefährlich sind, virtuell zu produzieren, ohne dass eine reale Person diese Aktionen tatsächlich ausführen muss. Aber auch Videokonferenzen und Telepresence-Applikationen können von fotorealistischen 3D-Charakteren profitieren, da diese die immersive Erfahrung von solchen Applikationen verstärken. Traditionelle Verfahren zum Rendern von fotorealistischen Bildern involvieren viele mühsame und zeitintensive Schritte, welche Expertenwissen vorraussetzen und zudem auch Rechenzeiten erreichen, die jenseits von Echtzeit sind. Diese Schritte beinhalten das Rigging und Skinning von virtuellen Charakteren, das Modellieren von Reflektions- und Materialeigenschaften sowie physikalisch basiertes Ray Tracing. Vor Kurzem haben Deep Learning-basierte Methoden versucht, die Rendering-Funktion von Daten zu lernen, was in Verfahren resultierte, die eine Nutzung durch Nicht-Experten ermöglicht. Allerdings basieren die meisten Methoden auf Synthese-Verfahren im 2D-Bildbereich und können daher keine 3D-Konsistenz garantieren. Darüber hinaus gelingt es den meisten Methoden auch nicht, bewegungs- und blickwinkelabhängige Effekte zu erzeugen. Daher präsentiert diese Thesis eine neue Methode und eine laufende Forschungsarbeit zum Thema Charakter-Synthese, die es erlauben, fotorealistische und kontrollierbare 3D-Charakteren synthetisch zu rendern, die nicht nur 3D-konsistent sind, sondern auch bewegungs- und blickwinkelabhängige Effekte modellieren und Echtzeit-Rechenzeiten ermöglichen. Dazu wird eine neuartige Grob-zu-Fein-Charakterrepräsentation für effiziente Bild-Synthese von Menschen vorgestellt, welche nur anhand von Multi-Kamera-Daten trainiert werden kann. Daneben wird gezeigt, wie diese explizite Geometrie-Repräsentation mit einer impliziten Oberflächendarstellung kombiniert werden kann, was eine bessere Synthese von geomtrischen Deformationen sowie Bildern ermöglicht.

ACKNOWLEDGMENTS

Working towards my Ph.D. over the last four years was an amazing and insightful time that I would not like to miss. I have had the chance to learn a lot from talented people, have grown beyond myself during this time, and have overcome many hurdles that seemed insurmountable. Along this journey, I have met many wonderful people to whom I owe a debt of gratitude.

I must start with Christian, the person who has influenced this journey the most: I would like to say thanks for the incredible supervision during my Ph.D. studies, from the initial offer to become a Ph.D. student in his research group, to the many inspiring meetings we had, his enthusiasm about research, and also his positive words when a project was rejected. For me, Christian is definitely a role model not only as a researcher but also as a person. I can only hope to become half the researcher that he is. That would make me very happy.

Next, I would like to thank my close collaborators, Michael, Weipeng, Gerard, and Lingjie, who, in various ways, helped me to realize my research projects over the past years. I enjoyed our discussions, tuning the writing to the limits, and the interactive coding sessions.

I would also like to thank the students and interns, Lan, Yuxiao, Yue, Linjie, and other collaborators I have been privileged to work with. They have done a great job, which has resulted in amazing projects.

Two people to whom I also owe special thanks are Sabine and Ellen. They do such a great job supporting the Ph.D. students. Without their help, my Ph.D. work would probably have failed due to administrative reasons such as filling out the Travel Expense Statement.

Further, I would like to thank our IT admins, Gereon, Jozef, and Hyengwoo, and the IST for their technical support. Whenever I required yet another GPU for my projects, they found a way to provide it.

Then, I would like to thank Hans-Peter and Tobias because it was they who offered me a HiWi position at the Max Planck Institute, where I then gained early experience in the role of a teacher when I supervised a local school project. This was also my first contact with the MPII and, so to say, the start of my membership in the Max Planck Society.

I would also like to thank the current and former GVV, D4, and D6 members, who have made my time at the Institute very entertaining and inspiring. I still remember all the interesting discussions we had during lunch and the bouldering sessions. Special thanks go to the master of calibration, Oleksandr, and my office mates Ikhsanul and Gereon, who have made the office hours a really joyful time.

A big thanks also go to Antonia, Sabrina, Ayush, Jiayi, Linjie, Lingjie, Ikhsanul, Yue, Edgar, Michael, Weipeng, Mohamed, Krista, and Soshi, who proofread this thesis.

I would also like to thank Hans-Peter Seidel and Adrian Hilton for being part of my thesis committee.

Last but not least, I would like to thank my wife, Antonia, and my family, who have absolutely supported me during my time as a Ph.D. student. I know it was sometimes not easy to get along with me, especially before deadlines, when I was having to work 24/7 and barely had time for anything else. Nonetheless, I had your unlimited support, love, and patience. I truly appreciate this.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

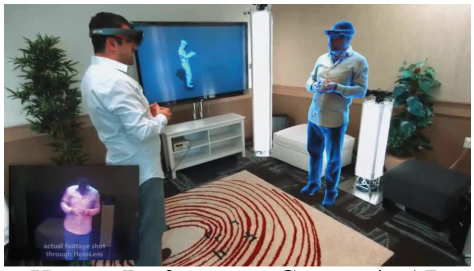
1.1 MOTIVATION

Most of the images one finds in the media, such as on the Internet or in textbooks and magazines, contain humans as the main point of attention. Thus, it is of enormous interest for industry, private people, and society to *analyze* and *synthesize* such *human-centered visual content* in an automated way.

Analyzing visual content like images or videos of humans typically means inferring a deeper understanding of a scene, such as the 3D pose or motion of the human or even its entire 3D surface just from the visual information (see Figure 1.1). In the literature, inferring the skeletal pose from images is also called motion capture, and inferring the entire time-varying geometry is called human performance capture. As motion capture makes it possible to recover the 3D skeletal pose simply from images, it is quite relevant for medical applications, namely to analyze the patient's recovery status and, if needed, to provide useful feedback for the therapist. But there are many more sports, entertainment, and surveillance applications where captured images of motions are used to analyze sports exercises' effectiveness, animate virtual characters, or detect suspicious activities, respectively. However, capturing skeletal motion alone does not suffice for many applications. One of these applications is a virtual try-on where the subject wants to receive interactive feedback about whether the clothing fits their body or not. Also, augmented and virtual reality applications prefer dense character tracking over pure skeleton capture since immersive effects can only be achieved when the 3D human characters look realistic from all aspects. Thus, besides the skeletal motion, a dense tracking of the entire surface, including the clothing, is also of great importance to reach the necessary level of realism.

While the above discussion mainly focused on analyzing the visual scene content, another aspect of human-centered Computer Vision is synthesizing photo-realistic humans from user-generated skeletal motions (see Figure 1.1). Having such a photoreal avatar would enable animation such that the avatar performs the desired actions, and it would also allow it to be rendered from an arbitrary virtual viewpoint. Such a setting would provide cinematic artists a simpler and more intuitive tool to create visual content, which is very time-consuming and tedious to produce when using the traditional Computer Graphics pipeline. For example, one can design a virtual double of an actor, insert photo-realistic actors into a fully virtual world, or synthesize

Motion Capture
(TheCaptury)



Human Performance Capture in AR
(Orts et al., 2016)

Figure 1.1: Left. Analyzing images and videos of humans allows the extraction of useful 3D information such as the skeletal pose, which can be used in medical applications, for example. Right. Recovering 3D models of humans from video data allows character synthesis in an augmented or virtual reality. © The respective copyright owners.

performances that would have been too dangerous for the actors on the film set. In fact, photo-realistic humans can also be advantageous for augmented and virtual reality setups to enhance either the real or the virtual world with digital doubles of real humans.

Currently, capturing the skeletal motion or the dense human surface relies on complicated hardware setups involving marker suits, depth cameras, or multiple cameras. Unfortunately, marker suits do not allow the capture of a person's everyday clothing, multiple cameras are expensive and require explicit calibration, and depth sensors cannot work in environments with bright sunlight and also consume a lot of power. Thus, none of the above are ideal solutions to the problem of human performance capture, and the use of these setups is inherently restricted to people who have expert knowledge and can afford such expensive equipment. To standardize human performance capture, one would ideally require only a single RGB camera, which everyone has on their smartphone or laptop. Moreover, the ideal method should capture the skeleton motion and the entire surface and its space-time-coherent deformation to create a more complete and realistic capture of the performance. This is quintessential for applications where both realism and an immersive experience are desirable.

For synthesizing virtual humans, the traditional computer graphics pipeline used for visual effects is very complicated, expensive, and time-consuming - in many aspects, from the character modeling to the rigging, skinning, and the rendering itself. Unlike this classical process, one would ideally like to have a very intuitive creation of the photo-real avatar, e.g., an artist would just have to define the motion. Then, the photo-real rendering is generated in an automated way. Moreover, the rendering process should be fast such that interactive editing is possible. Most importantly, the results should look photo-realistic, blending into real scenes without visual artifacts.

Despite great progress, the problem of monocular human performance capture, as well as an intuitive, fast, and photo-real character synthesis, are far from being solved and are accompanied by very deep technical challenges. The monocular human performance capture setup is inherently ambiguous due to the unknown scene depth, the highly articulated structure of human bodies, and the lack of visual information in the case of occlusions where body parts are not visible to the camera. Similarly, creating photo-realistic characters remains a difficult process and is far from being intuitive. The traditional graphics pipeline for realistic character creation involves many steps, such as scanning, rigging, skinning, and physically based rendering – the last step here being especially time-consuming, which makes interactive editing nearly impossible. Ideally, one would like to be able to create animatable human characters directly from video data.

To tackle the problem of unknown scene depth and the lack of visual information in occlusion, recent research has focused on monocular human performance capture and has tried to disambiguate the task by employing deep learning techniques and inverse kinematics fitting. Some methods regress low dimensional parameters of data-driven body models. However, such models typically cannot capture clothing; rather, they can only capture the naked human body, which is not ideal for AR/VR applications. Further, they achieve a plausible overlay on the input images, but the 3D performance is far from accurate. Other methods regress independent geometries for each frame, which lack temporal coherence, preventing them from being used in applications such as re-texturing. Only a few methods focus on jointly tracking a coherent geometry over time, and these methods are far from real-time performance. In addition, similarly as before, they suffer from an inaccurate 3D performance. For the synthesis of photo-real humans, recent monocular methods have partially replaced the tedious graphics pipeline with deep learning modules or texture retrieval techniques and learn the character appearance from video data. While these methods allow the user to control the synthesis process more intuitively, their results have various weaknesses as well; they cannot model view-dependent effects, they barely generalize to unseen motions, they cannot handle loose clothing, or they cannot run in real time.

To overcome these shortcomings, this thesis advances the state of the art in terms of monocular human performance capture and controllable character synthesis in several ways. In particular, this thesis presents the first real-time monocular human performance capture approach that tracks the skeletal pose and the space-time coherent deforming geometry of the entire human. Further, this thesis proposes a novel learning-based approach for monocular human performance capture that leverages multi-view supervision during training to improve the 3D accuracy in terms of skeletal pose and 3D surface deformation. Finally, this thesis introduces a real-time character synthesis approach

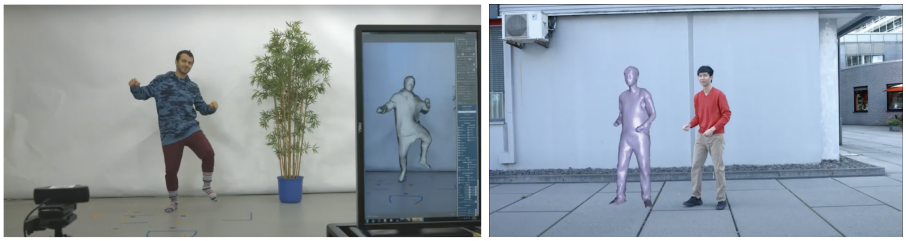


Figure 1.2: The proposed monocular human performance capture approaches allow one to extract the space-time coherent geometry of a human from a single color image (Chapter 4 and 5). Both methods advance the state of the art in terms of runtime performance and 3D accuracy, respectively.

that enables intuitive control and synthesis of photo-realistic characters solely learned from multi-view video data.

1.2 OVERVIEW

One goal of this thesis is to propose solutions that advance the state of the art and improve the task of monocular human performance capture, which targets the recovery of the dense 3D surface geometry of the entire human, including the clothing, from a single color image (see Figure 1.2). For many applications, it is desirable for the capture approach to run in real time in order to allow interactive feedback, which is, for example, required for tasks such as virtual try-on, texture augmentation, and applications in VR and AR. Key challenges in the monocular setting are the inherent ambiguities; self-occlusions and the generally unknown absolute depth can lead to strong ambiguities in terms of 3D pose and surface deformation. This thesis attempts to solve these challenges by introducing an efficient optimization-based tracking algorithm that jointly captures the 3D pose and surface deformation in real time. Further, a learning-based approach is proposed, demonstrating superior 3D performance in terms of pose and surface accuracy.

However, not only the analysis of human-centered image and video content is important, but also the capability to create novel content involving the synthesis of novel views and unseen motions (see Figure 1.3). Here, it is essential to ensure 3D consistency when changing the virtual viewpoint, to capture motion-dependent deformations and appearance changes, and to model view-dependent effects, such as specular reflections. In this thesis, new methods are presented that explicitly capture a motion- and view-dependent as well as 3D consistent geometry and appearance.

In Chapter 4, LiveCap is introduced, which is a monocular human performance capture approach that, for the first time, demonstrates

Habermann et al., 2021

Figure 1.3: The proposed human synthesis approach allows the photo-realistic rendering of novel views and motions of a given actor. This enables visual effects such as a person is fighting a virtual double (see right image).

real-time performance while also being able to recover dense surface deformations such as clothing wrinkles. The approach assumes that a rigged character template of the subject is given. Then, sparse and dense image cues are extracted from the individual video frames, and in the first stage, the pose is optimized to match the monocular observations. In a second stage, the non-rigid surface deformations are optimized using dense photometric energy terms starting with the posed template as initialization. Notably, all energy terms are efficiently solved in a data-parallel manner on the GPU using dedicated optimization techniques and novel algorithmic design choices all geared towards real-time performance.

In Chapter 5, DeepCap is introduced, which is a novel deep learning approach for monocular dense human performance capture. The proposed method is trained in a weakly supervised manner based on multi-view supervision, completely removing the need for training data with 3D ground truth annotations. This multi-view supervision has the advantage that at test time, monocular ambiguities such as occlusions and depth ambiguity can be resolved, which significantly improves the 3D accuracy compared to the state of the art.

As mentioned earlier, analyzing the human-centered image and video content is only one goal of this thesis. Another one is photo-realistic character synthesis. Therefore, in Chapter 6, a novel learning-based approach for video-based character synthesis is proposed. This method jointly models motion- and view-dependent surface deformation as well as appearance. In contrast to previous work, the explicit modeling of a deforming geometry allows for view-consistent 3D results, and the appearance recovers view- and motion-dependent effects in real time. To further improve the geometric accuracy and the synthesis quality, Chapter 6 also introduces an ongoing work for user-controlled 3D character synthesis using a combination of explicit and implicit geometry representations.

1.3 STRUCTURE

In the following, the contents of the individual chapters of this thesis are summarized:

- Chapter 1 motivates the topic of this thesis, provides an overview and structure of its content, and a summary of all the contributions made in the individual publications as well as a list of all the published works.
- Chapter 2 discusses works related to the proposed approaches that are presented in this thesis.
- Chapter 3 introduces the foundational concepts used later in the respective publications.
- Chapter 4 proposes the first method for real-time human performance capture, which requires only a single RGB camera and which can densely capture the dense human surface, including the clothing deformations.
- Chapter 5 presents a new learning-based method for monocular human performance capture that shows superior 3D reconstruction accuracy by leveraging weak multi-view supervision during training.
- Chapter 6 proposes a new learning-based method for motion-driven synthesis of photo-realistic human characters. This method jointly captures motion-dependent geometry as well as motion- and view-dependent dynamic appearance effects while running in real time. Moreover, an ongoing work is presented, which further improves the geometric detail and the synthesis quality of controllable and photo-realistic 3D characters by combining explicit and implicit geometry representations.
- Chapter 7 summarizes the insights that have been acquired in the proposed approaches and provides an outlook for future projects, for which the presented work will serve as a basis.

1.4 SUMMARY OF CONTRIBUTIONS

The contributions made in Chapter 4 (published as Habermann et al., 2019) are:

- The first real-time system for monocular human performance capture is presented. To achieve real-time performance, several new algorithmic concepts are presented to guarantee high-quality results under a tight real-time constraint.

- It is shown how to efficiently implement these design decisions by combining the compute power of two GPUs and the CPU in a pipelined architecture and how dense and sparse linear systems of equations can be efficiently optimized on the GPU.
- The approach is evaluated on a wide range of data where high-quality results are shown on an extensive new dataset of more than 20 minutes of video footage captured in 11 scenarios, containing different types of loose apparel and challenging motions.

Further, the new concepts presented in Chapter 5 (published as Habermann et al., 2020 and Habermann et al., 2021b) can be summarized as:

- A learning-based 3D human performance capture approach that jointly tracks the skeletal pose and the non-rigid surface deformations from monocular images.
- A new differentiable representation of deforming human surfaces that enables training from multi-view video footage directly.

Finally, the contributions in Chapter 6 (published as Habermann et al., 2021a) are:

- The first learning-based real-time approach that takes a motion and camera pose as input and predicts the motion-dependent surface deformation and motion- and view-dependent texture for the full human body. The approach is trained using weak 2D supervision only.
- A differentiable 3D character representation that can be trained from coarse to fine.
- A graph convolutional architecture allowing the formulation of the learning problem as a graph-to-graph translation task.
- A new benchmark dataset, called *DynaCap*, containing 5 actors captured with a dense multi-view system, which is publicly available for research.
- A combined explicit and implicit geometry representation is introduced, which further improves the geometric details and the synthesis quality of controllable and photo-realistic 3D characters.

1.5 PUBLICATIONS

The following lists peer-reviewed publications accepted at top-tier conferences and journals, and which are presented in this thesis:

- Marc Habermann et al. (2019). "LiveCap: Real-Time Human Performance Capture From Monocular Video." In: *ACM Transactions on Graphics (TOG)*. ACM
- Marc Habermann et al. (2020). "DeepCap: Monocular Human Performance Capture Using Weak Supervision." In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE **[CVPR 2020 Best Student Paper Honorable Mention]**
- Marc Habermann et al. (2021b). "A Deeper Look into DeepCap." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. IEEE
- Marc Habermann et al. (2021a). "Real-time Deep Dynamic Characters." In: *Proceedings of Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*. ACM

Further, I contributed to the following works, which are also published at top-tier conferences and journals; however, these are not part of this thesis:

- Marc Habermann et al. (2018). "NRST: Non-rigid Surface Tracking from Monocular Video." In: *Proceedings of the German Conference on Pattern Recognition (GCPR)*. Springer
- Lingjie Liu et al. (2019a). "Neural Rendering and Reenactment of Human Actor Videos." In: *ACM Transactions on Graphics (TOG)*. ACM
- Yuxiao Zhou et al. (2020). "Monocular Real-time Hand Shape and Motion Capture using Multi-modal Data." In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE
- Lan Xu et al. (2020). "EventCap: Monocular 3D Capture of High-Speed Human Motions using an Event Camera." In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE
- Lingjie Liu et al. (2020b). "Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation." In: *Transactions on Visualization and Computer Graphics (TVCG)*. IEEE
- Yuxiao Zhou et al. (2021). "Monocular Real-time Full Body Capture with Inter-part Correlations." In: *In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE

- Linjie Lyu et al. (2021). "Efficient and Differentiable Shadow Computation for Inverse Problems." In: *In Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE

- Lingjie Liu et al. (2021). "Neural Actor: Neural Free-view Synthesis of Human Actors with Pose Control." In: *ACM Transactions on Graphics (Proc. ACM SIGGRAPH Asia (conditionally accepted))*. ACM

Finally, I contributed to the following pre-published work, which is currently available on arXiv:

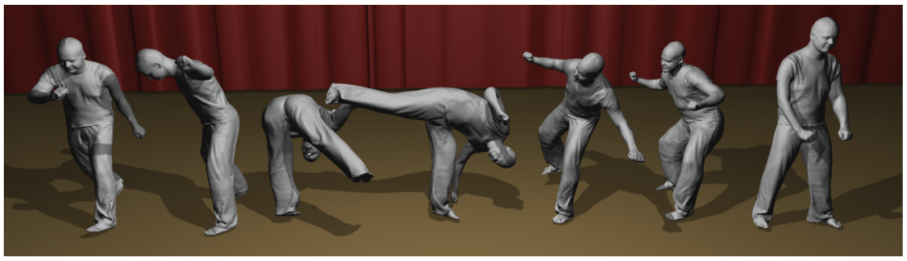
- Yue Li et al. (2020a). "Deep Physics-aware Inference of Cloth Deformation for Monocular Human Performance Capture." In: arXiv: 2011.12866 [cs.CV]

2 RELATED WORK

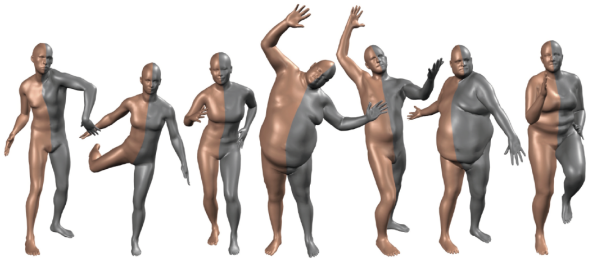
Human performance capture is a well-studied field in Computer Vision and Graphics, and many works were proposed in recent years. A key difference between these approaches is the type of input used. Thus, in the following, works are categorized into approaches that leverage multi-view imagery (Section 2.1), depth streams (Section 2.2), and, most related to the works presented in this thesis, single RGB images (Section 2.3). However, as mentioned earlier, capturing the performance is only one goal of this thesis, while synthesizing virtual humans under novel motions and camera views is the other. To this end, methods that aim to create video-based characters are discussed (Section 2.4). Neural and differentiable rendering are important components for the capture and synthesis of humans, and, hence, they are discussed in Section 2.5. As the realistic motion of cloth is important for capturing natural-looking human performances, methods introduced in this thesis are also capable of modeling motion-dependent deformations of the apparel; thus, learning-based cloth deformation works are reviewed in Section 2.6.

2.1 MULTI-VIEW BASED HUMAN PERFORMANCE CAPTURE

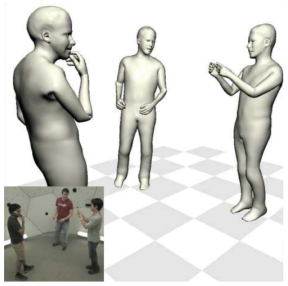
In the following, previous works that leverage multi-view imagery for reconstructing the dense and deforming human surface are reviewed (see also Figure 2.1). Many multi-view methods use stereo and shape from silhouette cues to capture the moving actor (Collet et al., 2015; Matusik et al., 2000; Starck and Hilton, 2007; Waschbüsch et al., 2005), or reconstruct the human via multi-view photometric stereo (Vlasic et al., 2009). Provided with sufficient images, some methods directly non-rigidly deform a subject specific template mesh (Cagniart et al., 2010; Carranza et al., 2003; De Aguiar et al., 2008) or a volumetric shape representation (Allain et al., 2015; Huang et al., 2016). Such methods are free-form and can potentially capture arbitrary shapes (Mustafa et al., 2016) as they do not incorporate any skeletal constraints. Such flexibility comes at the cost of robustness. To mitigate this, some methods incorporate a skeleton in the template to constrain the motion to be nearly articulated (Gall et al., 2009; Liu et al., 2011; Vlasic et al., 2008). This also enables off-line performance capture from a stereo pair of cameras (Wu et al., 2013). Some systems combine reconstruction and segmentation to improve results (Bray et al., 2006; Brox et al., 2010; Liu et al., 2011; Wu et al., 2012). Such methods typically require a high-resolution scan of the person as input. To side step scanning, a



de Aguiar et al., 2008



Loper et al., 2015



Joo et al., 2018

Figure 2.1: Top. Some works (Cagniart et al., 2010; Carranza et al., 2003; De Aguiar et al., 2008) reconstruct the dense and deforming surface from multi-view images. Bottom left. Other works (Anguelov et al., 2005; Loper et al., 2015) first build a statistical body model from thousands of scans. Bottom right. These body models are then tracked using multi-view constrains (Joo et al., 2018). © The respective copyright owners.

parametric body model can be employed. Early models were based on simple geometric primitives (Metaxas and Terzopoulos, 1993; Plänkers and Fua, 2001; Sigal et al., 2004; Sminchisescu and Triggs, 2003). Recent ones are more accurate, detailed, and are learned from thousands of scans (Anguelov et al., 2005; Hasler et al., 2010; Hesse et al., 2018; Kadlecek et al., 2016; Kim et al., 2017; Loper et al., 2015; Park and Hodgins, 2008; Pons-Moll et al., 2015). Capture approaches that use a statistical body model typically ignore clothing, treat it as noise (Balan et al., 2007), or explicitly estimate the shape under the apparel (Bălan and Black, 2008; Yang et al., 2016; Zhang et al., 2017). The offline human performance capture approach of Huang et al., 2017 fits the SMPL body model to 2D joint detections and silhouettes of the multi-view data. Some of the recent off-line approaches jointly track facial expressions (Joo et al., 2018) and hands (Joo et al., 2018; Romero et al., 2017). To capture the geometry of the actor beyond the body shape, an option is to non-rigidly deform the base model to fit a scan (Zhang et al., 2017) or a set of images (Rhodin et al., 2016). Recently, the approach of Pons-Moll et al., 2017 can jointly capture body shape, and clothing using separate meshes; very realistic results are achieved with this method, but it requires an expensive multi-view active stereo setup. Multi-view CNNs can map 2D images to 3D volumetric fields enabling reconstruction of a clothed human body at arbitrary resolution (Huang

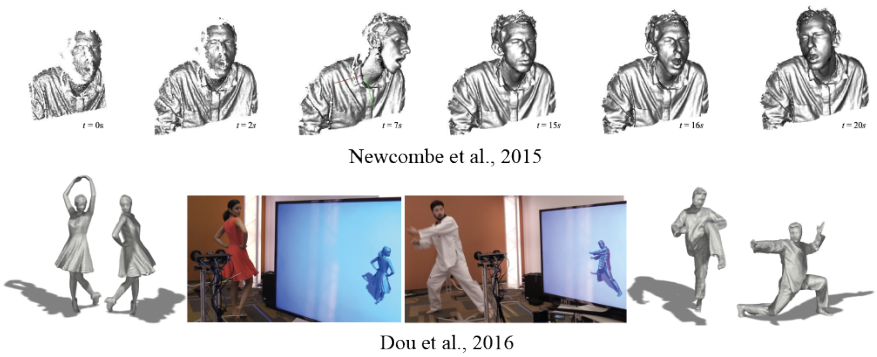


Figure 2.2: Top. Monocular depth-based methods (Guo et al., 2017; Newcombe et al., 2015) usually fuse unseen geometric details into a canonical volume, resulting in an improved reconstruction quality over time. Bottom. This concept was also extended to multiple sensors (Dou et al., 2016; Orts-Escolano et al., 2016), which allows a more robust tracking of faster performances and a higher quality. © The respective copyright owners.

et al., 2018). All the aforementioned approaches require multi-view setups and are not practical for consumer use. Furthermore, none of the methods runs at real-time frame rates.

2.2 DEPTH-BASED HUMAN PERFORMANCE CAPTURE

With the availability of affordable depth camera sensors, e.g., the Kinect, many depth-based methods emerged (see also Figure 2.2). Recent approaches that are based on a single depth camera, such as KinectFusion, enable the reconstruction of 3D rigid scenes (Izadi et al., 2011; Newcombe et al., 2011) and also appearance models (Zhou and Koltun, 2014) by incrementally fusing geometry in a canonical frame. DynamicFusion (Newcombe et al., 2015) generalized KinectFusion to capture dynamic non-rigid scenes. The approach alternates non-rigid registration of the incoming depth frames with updates to the incomplete geometry, which is constructed incrementally. Such template-free methods (Guo et al., 2017; Innmann et al., 2016; Newcombe et al., 2011; Slavcheva et al., 2017) are flexible but are limited to capturing slow and careful motions. One way to make fusion and tracking more robust is by using a combination of a high frame rate/low resolution and a low frame rate/high-resolution depth sensor (Guo et al., 2018), improved hardware and software components (Kowdle et al., 2018), multiple Kinects or similar depth sensors (Dou et al., 2017, 2016; Orts-Escolano et al., 2016; Ye et al., 2012; Zhang et al., 2014a), or multi-view data (Collet et al., 2015; Leroy et al., 2017; Prada et al., 2017) and registering new frames to a neighboring keyframe; such methods achieve impressive reconstructions, but do not register all frames to the same canonical template and require complicated capture setups. Another way to con-

strain the capture is to pre-scan the object or person to be tracked (De Aguiar et al., 2008; Ye et al., 2012; Zollhöfer et al., 2014), reducing the problem to tracking the non-rigid deformations. Constraining the motion to be articulated is also shown to increase robustness (Yu et al., 2017, 2018). Instead, HybridFusion (Zheng et al., 2018) additionally incorporates a sparse set of inertial measurement units. Some works use simple human shape or statistical body models (Bogo et al., 2015; Helten et al., 2013; Wei et al., 2012; Weiss et al., 2011; Ye and Yang, 2014; Zhang et al., 2014b,c), some of which exploit the temporal information to infer shape. Typically, a single shape and multiple poses are optimized to exploit the temporal information. Such approaches are limited to capture naked human shape or, at best, very tight clothing. Depth sensors are affordable and more practical than multi-view setups. Unfortunately, they have a high power consumption, do not work well under general illumination, and most media content is still in the format of 2D images and video. Furthermore, depth-based methods do not directly generalize to work with monocular video. In contrast, the methods presented in this thesis can work on the more popular RGB format and can work in outdoor conditions that are not suitable for depth-based methods.

2.3 MONOCULAR 3D POSE ESTIMATION AND HUMAN PERFORMANCE CAPTURE

Next, related works that only leverage a single RGB image are reviewed (see also Figure 2.3). Most methods to infer 3D human motion from monocular images are based on convolutional neural networks (CNNs) and leverage 2D joint detections and predict 3D joint pose in the form of stick figures (Popa et al., 2017; Rogez et al., 2017; Sun et al., 2017; Tome et al., 2017; Zhou et al., 2017). Tekin et al., 2016 directly predict the 3D body pose from a rectified spatio-temporal volume of input frames. The approach of Tekin et al., 2017 learns to optimally fuse 2D and 3D image cues. These approaches do not capture the dense deforming shape. One work within this thesis (Chapter 4) also leverages a recent CNN-based 3D pose estimation method (Mehta et al., 2017), but it is only employed to regularize the skeletal motion estimation. Some works fit a (statistical) body surface model to images using substantial manual interaction (Guan et al., 2009; Jain et al., 2010; Rogge et al., 2014; Zhou et al., 2010) typically for the task of image manipulation. Shape and clothing can also be recovered (Chen et al., 2013; Guo et al., 2012), but the user needs to click points in the image, select the clothing types from a database, and dynamics are not captured. Instead of clicked points, Kraevoy et al., 2009 propose to obtain the shape from contour drawings. With the advance of 2D joint detections, some works (Bogo et al., 2016; Kanazawa et al., 2018; Kolotouros et al., 2019; Lassner et al., 2017) fit a 3D body model (Loper

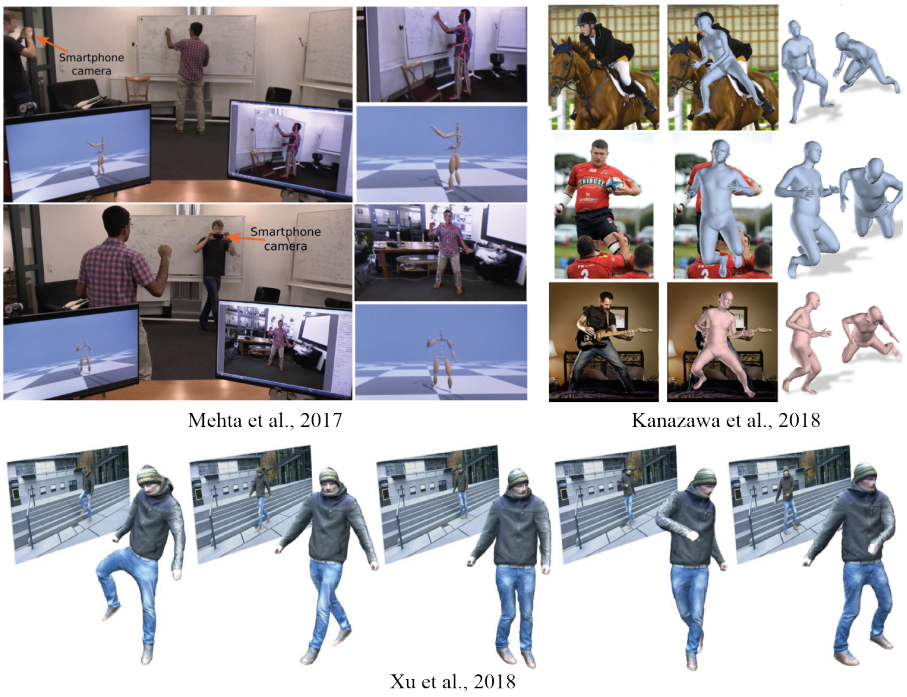


Figure 2.3: Top left. Some works (Mehta et al., 2017; Sun et al., 2017) predict a sparse 3D skeletal pose from single images. Top right. Others (Kanazawa et al., 2018; Kolotouros et al., 2019) jointly predict the 3D pose and shape of the naked human leveraging a statistical body model. Bottom. Further, there are template-based approaches (Xu et al., 2018), which optimize for dense 3D surface deformations using foreground masks. © The respective copyright owners.

et al., 2015) to them; since only model parameters are optimized, the results are constrained to the shape space. An alternative is to regress model parameters directly (Kanazawa et al., 2018, 2019; Pavlakos et al., 2018) or directly regressing a coarse volumetric body shape (Varol et al., 2018). Correspondences from pixels of an input image to surface points on the SMPL body model can also be directly regressed (Güler et al., 2018). Some works also jointly regress the skeletal body pose with facial expressions and hand gestures (Pavlakos et al., 2019; Xiang et al., 2019; Zhou et al., 2021). Capturing 3D non-rigid deformations from monocular video is very hard. In the domain of non-rigid structure from motion, model-free methods using rigidity and temporal smoothness priors can capture coarse 3D models of simple motions and medium-scale deformations (Garg et al., 2013; Russell et al., 2014). Some methods (Bartoli et al., 2015; Salzmann and Fua, 2011; Yu et al., 2015) can non-rigidly track simple shapes and motions by off-line template fitting, but they were not shown to handle highly articulated fast body motions, including clothing. Specifically for faces, monocular performance capture methods were presented (Cao et al., 2015; Garrido

et al., 2016). However, monocular full-body capture faces additional challenges due to more frequent (self-)occlusions and much more complex and diverse clothing as well as appearance. The pioneering work of Xu et al., 2018 shows for the first time that 3D performance capture of the human body, including the non-rigid deformation of clothing from monocular video, can be achieved. Its space-time formulation can resolve difficult self-occluded poses at the expense of temporally oversmoothing the actual motion. It is also challenged by starkly non-rigidly moving clothing. Recently, MonoClothCap (Xiang et al., 2020) removes the need for a personalized template but instead deforms the SMPL model while capturing the performance of the actor. Both methods report a runtime of over 1 minute per frame, which is impractical for many applications such as virtual try-on, gaming, or virtual teleportation. Reducing the processing time without compromising accuracy introduces challenges in the formulation and implementation of model-based performance capture, which is addressed in this thesis. To this end, for the first time, a real-time full-body performance capture system (Chapter 4) is presented that only requires a monocular video as input. It is shown that it comes close in accuracy to the best off-line monocular and even multi-view methods while being orders of magnitude faster. Moreover, current monocular methods suffer from the inherent depth ambiguity and occlusions resulting in limited 3D performance. To overcome these limitations, this thesis also presents a monocular human performance capture approach (Chapter 5) that reports state of the art 3D accuracy in terms of sparse 3D pose as well as dense 3D surface deformation by leveraging multi-view supervision and a dedicated coarse-to-fine regression strategy, which can be trained entirely weakly supervised.

2.4 VIDEO-BASED CHARACTERS

Previous work in the field of video-based characters aims at creating photo-realistic renderings of controllable virtual avatars under unseen motions and viewpoints (see also Figure 2.4). Classical methods attempt to achieve this by synthesizing textures on surface meshes and/or employing image synthesis techniques in 2D space. Some works (Carranza et al., 2003; Collet et al., 2015; Hilsmann et al., 2020; Li et al., 2014; Zitnick et al., 2004) focus on achieving free-viewpoint replay from multi-view videos with or without 3D proxies, however, they are not able to produce new motions for human characters. The approach of Stoll et al., 2010 incorporates a physically based cloth model to reconstruct a rigged fully-animatable character in loose cloths from multi-view videos, but it can only synthesize a fixed static texture for different poses. To render the character with dynamic textures in new poses from arbitrary viewpoints, Xu et al., 2011 propose a method that first retrieves the most similar poses and viewpoints in a pre-captured

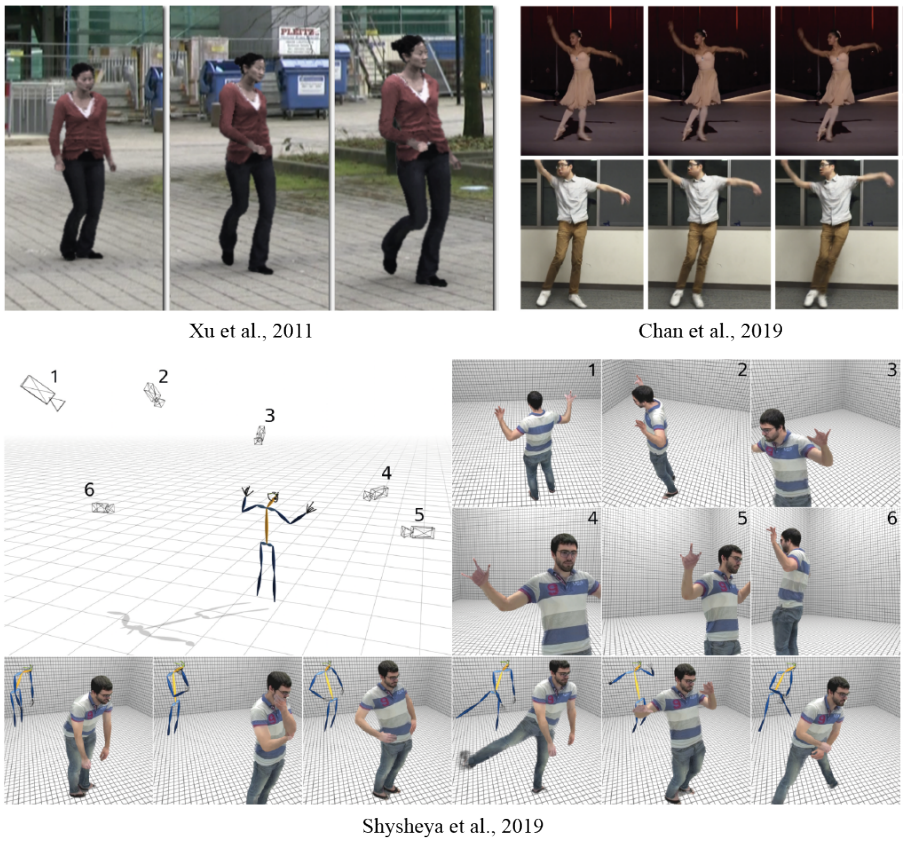


Figure 2.4: Top left. Some works (Casas et al., 2014; Xu et al., 2011) apply texture retrieval techniques from a multi-view database of the actor's performance. Top right. Recent learning-based approaches (Chan et al., 2019; Liu et al., 2020b; Pumarola et al., 2018) only leverage a single camera to synthesize unseen actor motions. Bottom. Other learning-based approaches (Shysheya et al., 2019) leverage multi-view data and learn a 2D texture that is used to create the final rendering. © The respective copyright owners.

database and then applies retrieval-based texture synthesis. However, their method takes several seconds per frame and thus cannot support interactive character animation. Casas et al., 2014 and Volino et al., 2014 compute a temporally coherent layered representation of appearance in texture space to achieve interactive speed, but the synthesis quality is limited due to the coarse geometric proxy. Most of the traditional methods for free-viewpoint rendering of video-based characters fall either short in terms of generalization to new poses and/or suffer from a high runtime and/or a limited synthesis quality.

More recent works employ neural networks to close the gap between rendered virtual characters and real captured images. While some approaches have shown convincing results for the facial area (Kim et al., 2018a; Lombardi et al., 2018), creating photo-real images of the entire human is still a challenge. Most of the methods, which target synthesizing entire humans, learn an image-to-image mapping

from renderings of a skeleton (Chan et al., 2019; Esser et al., 2018; Pumarola et al., 2018; Si et al., 2018), depth map (Martin-Brualla et al., 2018), dense mesh (Liu et al., 2020b, 2019a; Sarkar et al., 2020; Wang et al., 2018a) or joint position heatmaps (Aberman et al., 2019), to real images. Among these approaches, the most related work (Liu et al., 2020b) achieves better temporally-coherent dynamic textures by first learning fine-scale details in texture space and then translating the rendered mesh with dynamic textures into realistic imagery. While only requiring a single camera, these methods only demonstrate the rendering from a fixed camera position, while the approach proposed within this thesis (Chapter 6) works well for arbitrary viewpoints and also models the view-dependent appearance effects. Further, these methods heavily rely on an image-to-image translation network to augment the realism. However, this refinement simply applied in 2D image space leads to missing limbs and other artifacts in their results. In contrast, the approach presented within this thesis does not require any refinement in 2D image space but explicitly generates high-quality view- and motion-dependent geometry and texture for rendering to avoid such kind of artifacts. Textured Neural Avatars (Shysheya et al., 2019) (TNA) also assumes multi-view imagery is given during training. However, TNA can neither synthesize motion- and view-dependent dynamic textures nor predict the dense 3D surface. This thesis proposes a method and ongoing work that can predict motion-dependent deformations on surface geometry as well as dynamic textures from a given pose sequence and camera view leading to video-realistic renderings.

2.5 NEURAL AND DIFFERENTIABLE RENDERING

Differentiable and neural rendering bridges the gap between 2D supervision and unknown 3D scene parameters that one wants to learn or optimize (see also Figure 2.5). Thus, differentiable rendering allows one to train deep architectures that learn the 3D parameters of a scene, solely using 2D images for supervision. OpenDR (Loper and Black, 2014) first introduces an approximate differentiable renderer by representing a pixel as a linear combination of neighboring pixels and calculating pixel derivatives using differential filters. Kato et al., 2018 propose a 3D mesh renderer that is differentiable up to the visibility assumed to be constant during one gradient step. Liu et al., 2019b differentiate through the visibility function and replace the z-buffer-based triangle selection with a probabilistic approach which assigns each pixel to all faces of a mesh. DIB-R (Chen et al., 2019) proposes to compute gradients analytically for all pixels in an image by representing foreground rasterization as a weighted interpolation of a face's vertex attributes and representing background rasterization as a distance-based aggregation of global face information. SDFDiff (Jiang

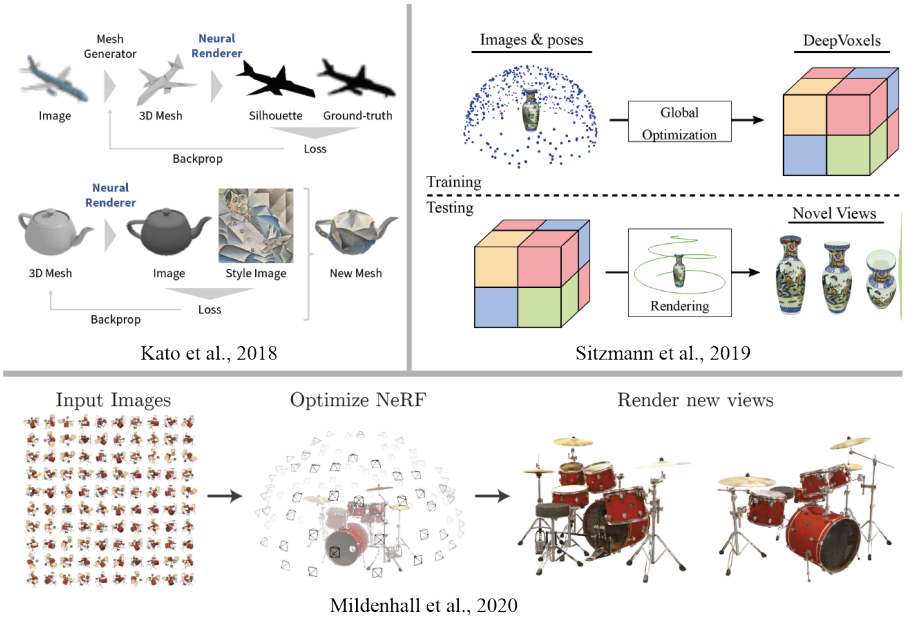


Figure 2.5: Top left. Some works (Kato et al., 2018; Loper and Black, 2014) make the rasterization process differentiable, allowing one to densely supervise scene properties using a photometric consistency loss between the rendering and the real image. Top right. Other approaches (Sitzmann et al., 2019a) focus on the learnable scene representation itself, such as a voxel grid where deep features are attached. Bottom. Alternatively, neural radiance fields (Mildenhall et al., 2020) model the scene as a volume, and volume rendering is applied to retrieve novel views of a static scene. © The respective copyright owners.

et al., 2020) introduces a differentiable renderer based on ray-casting signed distance functions. The implementation of differentiable rendering presented in this thesis (Chapter 6) follows the one of Kato et al., 2018 where the surface is modeled as non-transparent, and, thus, the visibility is non-differentiable. This is preferable for capturing humans as treating the human body and clothing as transparent would lead to wrong surface deformations and blurry dynamic textures.

Unlike differentiable rendering, neural rendering makes almost no assumptions about the physical model and uses neural networks to learn the rendering process from data to synthesize photo-realistic images. Some neural rendering methods (Aberman et al., 2019; Chan et al., 2019; Kim et al., 2018b; Liu et al., 2020b, 2019a; Ma et al., 2017, 2018; Martin-Brualla et al., 2018; Pumarola et al., 2018; Sarkar et al., 2020; Shysheya et al., 2019; Siarohin et al., 2018; Thies et al., 2019; Yoon et al., 2020) employ image-to-image translation networks (Isola et al., 2017a; Wang et al., 2018a,b) to augment the quality of the rendering. However, most of these methods suffer from view and/or temporal inconsistency. To enforce view and temporal consistency, some attempts were made to learn scene representations for novel view synthesis

from 2D images. Although this kind of methods achieve impressive renderings on static scenes (Liu et al., 2020a; Mildenhall et al., 2020; Sitzmann et al., 2019a,b; Zhang et al., 2020) and dynamic scenes for playback or implicit interpolation (Li et al., 2020b; Lombardi et al., 2019; Park et al., 2020; Peng et al., 2021; Pumarola et al., 2021; Raj et al., 2021; Tretschk et al., 2020; Wang et al., 2020b; Xian et al., 2020; Zhang et al., 2020) and faces (Gafni et al., 2021), it is not straightforward to extend these methods to synthesize full body human images with explicit pose control. Instead, the approach and the ongoing work presented in this thesis (Chapter 6) can achieve video-realistic renderings of the full human body with motion- and view-dependent dynamic textures for arbitrary body poses *and* camera views.

2.6 LEARNING-BASED CLOTH DEFORMATION

Modeling clothing and its deformations from images is also a widely studied field, and many works were proposed (see Figure 2.6). Synthesizing realistic cloth deformations with physics-based simulation has been extensively explored (Choi and Ko, 2005; Liang et al., 2019; Narain et al., 2012; Nealen et al., 2005; Su et al., 2020; Tang et al., 2018; Tao et al., 2019). They employ either continuum mechanics principles followed by finite element discretization or physically consistent models. However, they are computationally expensive and often require manual parameter tuning. To address this issue, some methods (Feng et al., 2010; Guan et al., 2012; Hahn et al., 2014; Kim and Vendrovsky, 2008; Wang et al., 2010; Xu et al., 2014; Zurdo et al., 2013) model cloth deformations as a function of the underlying skeletal pose and/or the shape of the person and learn the function from data.

With the development of deep learning, skinning-based deformations can be improved (Bailey et al., 2018) over the traditional methods like linear blend skinning (Magnenat-Thalmann et al., 1988) or dual quaternion skinning (Kavan et al., 2007). Other works go beyond skinning-based deformations and incorporate deep learning for predicting cloth deformations and learn garment deformations from the body pose and/or shape. Some works (Alldieck et al., 2019a, 2018a,b; Bhatnagar et al., 2019; Jin et al., 2020; Pons-Moll et al., 2017) generate per-vertex displacements over a parametric human model to capture the garment deformations. While this is an efficient representation, it only works well for tight clothes such as pants and shirts. Instead of such a discrete template mesh, some approaches (Saito et al., 2019, 2020) regress the body and the clothing geometry using an implicit surface representation, combine a coarse-scale volumetric reconstruction with a refinement network to add high-frequency details (Zheng et al., 2019), or use a multi-view silhouette representation (Natsume et al., 2019). Gundogdu et al., 2019 use neural networks to extract garment features at varying levels of detail (i.e., point-wise, patch-wise,

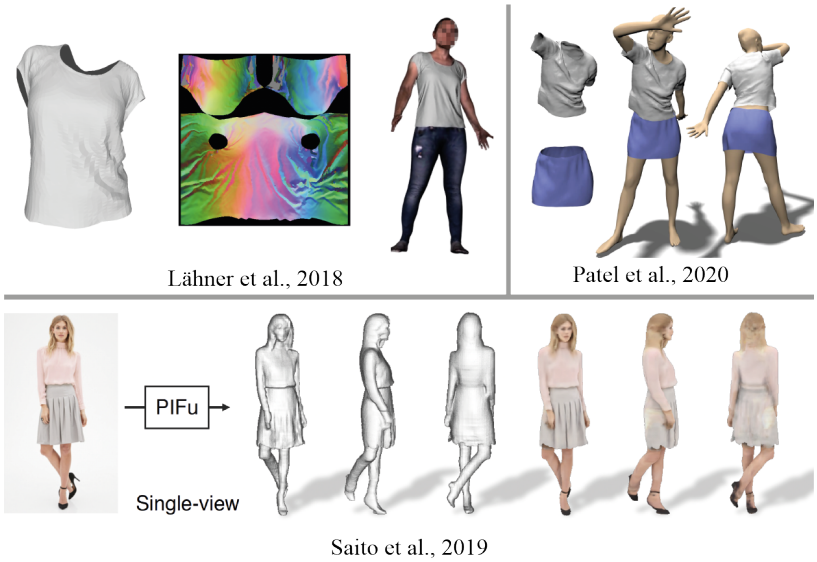


Figure 2.6: Top left. Some works (Lähner et al., 2018) decompose the clothing geometry into an explicit mesh component, and for modeling finer details, they leverage a 2D texture representation in addition. Top right. Others (Patel et al., 2020) model the clothing deformations entirely with a mesh-based representation but decompose the deformations into low and high frequencies. Bottom. A completely different approach to modeling clothing is to model the apparel as an implicit surface which is defined by the zero crossings of a (signed) distance function (Saito et al., 2019). © The respective copyright owners.

and global features). Patel et al., 2020 decompose the deformation into a high frequency and a low-frequency component. While the low-frequency component is predicted from pose, shape, and style of garment geometry with an MLP, the high-frequency component is generated with a mixture of shape-style specific pose models. Related to that, Choi et al., 2020 predict the geometry of the naked human from coarse to fine given the skeletal pose. Santesteban et al., 2019 separate the global coarse garment fit, due to body shape, from local detailed garment wrinkles, due to both body pose dynamics and shape. Other methods (Lähner et al., 2018; Zhang et al., 2021) recover fine garment wrinkles for high-quality renderings or 3D modeling by augmenting a low-resolution normal map of a garment with high-frequency details using GANs. Zhi et al., 2020 also reconstruct albedo textures and refine a coarse geometry obtained from RGB-D data. The method proposed in this thesis (Chapter 6) factors cloth deformation into low-frequency large deformations represented by an embedded graph and high-frequency fine wrinkles modeled by per-vertex displacements or a mesh-guided implicit surface representation, which enables the synthesis of deformations for any type of clothing, including also loose clothes. In contrast to the above methods, the proposed approach predicts not only geometric deformations but also a dynamic texture map that allows one to render photo-realistic controllable characters.

3

PREREQUISITES

Humans have a highly articulated structure with their arms, legs, and head. Thus, it is particularly hard to capture them when starting from the finest level, e.g., tracking the surface deformation directly on vertex level. Fortunately, humans have a piecewise rigid structure, the *skeleton*, which can describe coarse deformations in a lower-dimensional parameter space. Figure 3.1 shows a detailed 3D model of the anatomical structure of the human skeleton, but in visual computing, coarser approximations to the real human skeleton (see Figure 3.2) are more commonly used, and also this thesis leverages such a coarser version. Importantly, skeletons have a rather low number of degrees of freedom attached to their joints, enabling the control of the mesh with only a few parameters compared to directly editing single vertices. The works proposed in this thesis leverage this fact and model deformations of the human as a hierarchy of representations with the skeleton at the lowest level. Thus, in the following sections, the necessary prerequisites for skeleton-based deformation are described starting with how a skeleton can be animated (Section 3.1) to how the final mesh can be deformed according to the skeleton pose (Section 3.2).

3.1 KINEMATIC CHAIN

The human skeleton can be defined as a tree-like structure or graph, with the root usually being one of the spine joints (see Figure 3.2). Then, bones connect the joints with other joints building the tree-like structure, which ends in the end effectors, i.e., fingertips, toes, and head. When looking at the two ends of a bone, one can define the root-oriented side (ROS), which is the end of the bone closer to the root joint when traversing the previous bones, and the non-root-oriented side (NROS), which is defined vice versa. Importantly, at a joint, there can only be one

Figure 3.1: A detailed 3D model of the human skeleton depicting joints and bones, which form a kinematic structure. Figure from Kadlecek et al., 2016.

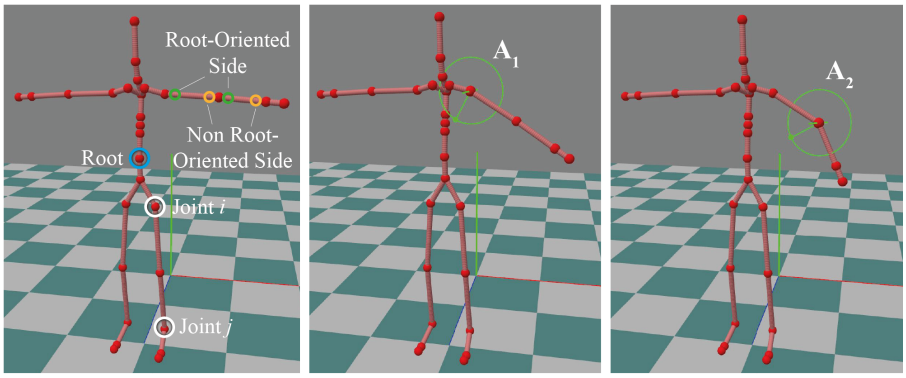


Figure 3.2: A visualization of the human skeleton structure, including the root joint, the (non) root-oriented sides of the bones, and the joints j and i. Here, i is a parent joint of j, and j is a child joint of joint i. Individual transforms can be applied along the kinematic chain, i.e., transforms on the shoulder and elbow joints, to finally pose the human's left hand.

non-root-oriented side but multiple root-oriented ones for a tree-like structure. Another important property is that the paths, which can be traversed, are uni-directional. Thus, when taking a step from a ROS to NROS, all consecutive steps have to go from ROS to NROS or vice versa. This property is important as it ensures that when one wants to traverse the tree from one joint to another one, there exists either a single path or no path at all. Further, for each joint, there exists a unique path to the root. This consideration leads to another internal relation between joints, which is the notion of a parent joint and a child joint. Here, a joint i is called a parent joint with respect to another joint j if and only if the path from j to the root passes i. Likewise, a joint j is called a child joint of joint i if and only if there exists a path, which connects i and j and where the distance to the root is larger for j compared to the distance between the root and i. This relationship finally defines a so-called *kinematic chain* (Reuleaux, 1875), which is the set of all joints k that are contained in the path from joint i to joint j.

Next, it is explained how the articulated motion of a skeleton structure can be controlled. To this end, a single or multiple degrees of freedom (DoF) can be assigned to a joint i. More precisely, the DoF is a rotation angle θ where the center of rotation is the center of the joint i and the rotation axis is locally defined and attached to the respective joint. The local rotation angle θ defines a rigid transform $\mathbf{A}_j \in SE(3)$ and the mapping from angles to rotation matrices depends on the angle representation used, i.e., Euler angles, dual quaternions, or axis angles. Independent of the type of rotation representation, the rigid transform can be generally defined as a function

$$a_j(\theta) = \mathbf{A}_j, \tag{3.1}$$

which takes the angle $\theta \in \mathbb{R}$ and returns the rigid transform $\mathbf{A}_j \in \mathbb{R}^{4\times4}$. A skeleton can have multiple DoFs, which define the *pose* $\boldsymbol{\theta} \in \mathbb{R}^D$ of the skeleton where D is the total number of DoFs. To determine the transform of a joint j, all transforms along the kinematic chain (starting from the joint j itself) to the root have to be consecutively applied, which can be written as the recursive function

$$\mathbf{A}^g_{root} = \mathbf{B}_{origin,j} \tag{3.2}$$

$$\mathbf{A}^g_j = \mathbf{A}^g_{par(j)} \mathbf{B}_{par(j),j} a_j(\boldsymbol{\theta}_j) \tag{3.3}$$

where $par(\cdot)$ denotes the parent of joint j. Here, $\mathbf{B}_{par(j),j} \in \mathbb{R}^{4\times4}$ is the transform defined by the local offset vector between the parent joint of j and joint j itself. Further, $\mathbf{B}_{origin,j} \in \mathbb{R}^{4\times4}$ is a transform defined by the offset between the root and the origin of the global coordinate system. Similarly, one can obtain the global position $\mathbf{p}^g_j \in \mathbb{R}^3$ of joint j as the translational component of $\mathbf{A}^g_j$. In addition to these local rotations, one also wants to move and rotate the entire skeleton in 3D space. Thus, additional 6 DoFs are usually attached to the root joint, 3 for the 3D translation $\mathbf{t} \in \mathbb{R}^3$ and 3 for the 3D rotation $\boldsymbol{\alpha} \in \mathbb{R}^3$. Altogether, the forward kinematics function can be defined as

$$f(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{t}) = \mathbf{P}^g, \tag{3.4}$$

which takes all DoFs as input and returns the matrix $\mathbf{P}^g \in \mathbb{R}^{J\times3}$ where row j contains the position of joint j in global space after applying the pose according to Equation 3.1, 3.2, and 3.3. Here, J denotes the total number of joints. It is important to note that this function is differentiable with respect to the joint angles, which allows backpropagation from the global 3D joint positions to the angles, which is essential for the inverse kinematics task solved in the later chapters.

So far, the skeleton can be posed by modifying the DoFs. However, instead of only posing the skeleton, it is important to pose a dense 3D mesh for most Computer Graphics and Vision applications in order to enhance the visual appearance. To this end, the next section provides the background for applying the skeletal pose to a mesh, which is referred to as *rigging* and *skinning*.

3.2 RIGGING AND SKINNING

To animate a mesh based on an underlying skeleton, the mesh has to be attached to the skeleton so that transformations along the kinematic chain are propagated to the mesh itself. Importantly, the kinematic transforms must be only locally applied to the mesh, e.g., a transform on the elbow does not influence mesh vertices around the knees. This can be achieved by so-called skinning weights. Assuming the skeleton contains K joints and the mesh has N vertices, the skinning weight

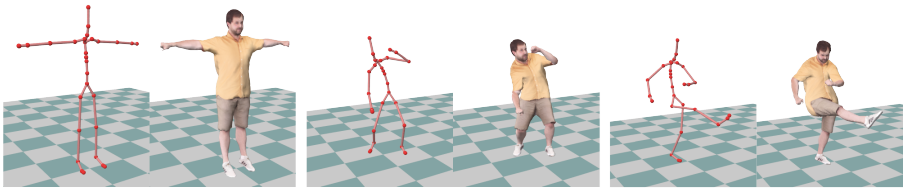


Figure 3.3: Visualization of Dual Quaternion Skinning. From left to right. The skeleton and mesh in the canonical pose. Different skeletal poses and the respective posed meshes using Dual Quaternion Skinning.

matrix $\mathbf{S}$ has a dimensionality of $N \times K$. Here, $\mathbf{S}$ has to satisfy certain properties such as all entries are non-negative, and the rows have to sum up to one. The intuition behind this is that the value $\mathbf{S}_{i,j}$ defines how much vertex i is influenced by the joint transformation $\mathbf{A}_j^g$. Most of these skinning weights are zero, with non-zero regions around their local influence zones. The process of aligning the sparse skeleton with a mesh in canonical pose and deriving those skinning weights is also called rigging, which can be either performed manually or automatically with 3D modeling software. However, for more details, it is, for example, referred to the work of Baran and Popović, 2007.

Given the skinning weights, one could compute the weighted sum of joint transformations for each vertex using the skinning weights, which is also called Linear Blend Skinning (LBS). However, as pointed out by Kavan et al., 2007, linear combinations of rotation matrices do not necessarily lead to a valid rotation matrix, such that so-called candy wrapper artifacts can arise. Thus, instead of LBS, all presented works in this thesis leverage Dual Quaternion Skinning (DQS) (Kavan et al., 2007), which is explained in more detail in the following. The idea of DQS is to convert the global joint transformation $\mathbf{A}_j^g$ of joint j to a dual quaternion $\mathbf{q}_j \in \mathbb{R}^8$, which is denoted as

$$\mathbf{q}_j = DQ(\mathbf{A}_j^g) \tag{3.5}$$

where $DQ(\cdot)$ is the conversion function. Now, to pose a vertex $\mathbf{v}_i$ the dual quaternions can be linearly blended according to

$$\mathbf{v}_i^g = DQ_{\mathrm{R}}^{-1}\left(\frac{\sum_{j=0}^{K}\mathbf{S}_{i,j}\mathbf{q}_j}{\|\sum_{j=0}^{K}\mathbf{S}_{i,j}\mathbf{q}_j\|}\right)\mathbf{v}_i + DQ_{\mathrm{T}}^{-1}\left(\frac{\sum_{j=0}^{K}\mathbf{S}_{i,j}\mathbf{q}_j}{\|\sum_{j=0}^{K}\mathbf{S}_{i,j}\mathbf{q}_j\|}\right) \tag{3.6}$$

where one has to note that the blending of the individual DQs results in a normalized DQ, which implies that it can be converted back to a *valid* rigid transform. Here, $DQ_{\mathrm{R}}^{-1}(\cdot)$ is a function that converts the rotational part of the DQ back into a valid rotation matrix. Similarly, $DQ_{\mathrm{T}}^{-1}(\cdot)$ is the function that converts the translational part of the DQ back into a translation vector. Important to note is that the only

variables here are the DoFs of the underlying skeleton, and moreover, the skinning process is fully differentiable. This is particularly helpful for inverse kinematics problems where losses and energy terms directly act on the vertices, e.g., the posed character should match a target point cloud. Due to the differentiability of the skinning, the supervision can be directly backpropagated into the pose variables of the skeleton.

As all the necessary tools for dense 3D character posing are introduced, the next chapter of this thesis presents the first *real-time* approach for dense monocular human performance capture.

4

LIVECAP: REAL-TIME HUMAN PERFORMANCE CAPTURE FROM MONOCULAR VIDEO

This chapter presents the first real-time human performance capture approach (published as Habermann et al., 2019) that reconstructs dense, space-time coherent deforming geometry of entire humans in general everyday clothing from just a single RGB video. A novel two-stage analysis-by-synthesis optimization is proposed whose formulation and implementation are designed for high performance. In the first stage, a skinned template model is jointly fitted to a background-subtracted input video, 2D and 3D skeleton joint positions found using a deep neural network, and a set of sparse facial landmark detections. In the second stage, dense non-rigid 3D deformations of skin and even loose apparel are captured based on a novel real-time capable algorithm for non-rigid tracking using dense photometric and silhouette constraints. The novel energy formulation leverages automatically identified material regions on the template to model the differing non-rigid deformation behavior of skin and apparel. The two resulting non-linear optimization problems are solved per frame with specially-tailored data-parallel Gauss-Newton solvers. In order to achieve a real-time performance of over 25Hz, a pipelined parallel architecture is designed which uses the CPU and two commodity GPUs. The proposed method is the first real-time monocular approach for full-body performance capture and yields comparable accuracy with off-line performance capture techniques while being orders of magnitude faster.

4.1 INTRODUCTION

Dynamic models of virtual human actors are key elements of modern visual effects for movies and games, and they are invaluable for believable, immersive virtual and augmented reality, telepresence, as well as 3D and free-viewpoint video. Such virtual human characters ideally feature high-quality, space-time coherent dense models of shape, motion, and deformation, as well as appearance of people, irrespective of physique or clothing style. Creating such models at high fidelity often requires many months of work of talented artists. To simplify the process, marker-less performance capture methods were researched to reconstruct at least parts of such models from camera recordings of real humans in motion.

Existing multi-camera methods are capable of capturing human models at very good quality, but they often need dense arrays of video

Figure 4.1: This chapter proposes the first real-time human performance capture approach that reconstructs dense, space-time coherent deforming geometry of people in their loose everyday clothing from just a single monocular RGB stream, e.g., captured by a webcam.

or depth cameras and controlled studios, struggle with complex deformations, and need pre-captured templates. Only a few multi-view methods achieve real-time performance, but no real-time method for single RGB performance capture exists. Many applications in interactive VR and AR, gaming, virtual try-on (Hilsmann and Eisert, 2009; Pons-Moll et al., 2017; Sekine et al., 2014), pre-visualization for visual effects, 3DTV or telepresence (Orts-Escolano et al., 2016) critically depend on real-time performance capture. The use of complex camera arrays and studios restricted to indoor scenes presents a practical barrier to these applications. In daily use, systems should ideally require only one camera and work outdoors.

Under these requirements, performance capture becomes a much harder and much more underconstrained problem. Some methods have approached this challenge by using multiple (Collet et al., 2015; Dou et al., 2016; Wang et al., 2016) or a single low-cost consumer-grade depth (RGB-D) (Newcombe et al., 2015; Yu et al., 2017) camera for dense non-rigid deformation tracking. While these methods are a significant step forward, RGB-D cameras are not as cheap and ubiquitous as color cameras, often have a limited capture range, do not work well under bright sunlight, and have limited resolution. Real-time human performance capture with a single color camera would therefore greatly enhance and simplify performance capture and further democratize its use, in particular in the aforementioned interactive applications of ever-increasing importance. However, dense real-time reconstruction from one color view is even harder, and so today's best monocular methods only capture very coarse models, such as bone skeletons (Mehta et al., 2017; Sun et al., 2017), or the naked human body (Kanazawa et al., 2018).

In this chapter, the first real-time human performance capture method is proposed that reconstructs dense, space-time coherent deforming geometry of people in their loose everyday clothing from a single video camera (see Figure 4.1). In a pre-processing step, the method builds a rigged surface and appearance template from a short video of the person in a static pose, on which regions of skin and pieces of apparel are automatically identified using a new multi-view segmentation that leverages deep learning. The template is fitted to the

video sequence in a new coarse-to-fine two-stage optimization, whose problem formulation and implementation are rigorously designed for best accuracy at real-time performance. In its first stage, the new real-time skeleton pose optimizer fits the skinned template to (1) 2D and 3D skeleton joint positions found with a CNN, to (2) sparse detected facial landmarks, and (3) to the foreground silhouette.

In a second stage, dense non-rigid 3D deformations of even loose apparel are captured. To this end, a novel real-time capable algorithm for non-rigid analysis-by-synthesis tracking from monocular RGB data is proposed. It minimizes a template-to-image alignment energy by jointly considering distance-field-based silhouette alignment, dense photometric alignment, and spatial and temporal regularizers, all designed for real-time performance. The energy formulation leverages the shape template segmentation labels (obtained in the pre-processing stage) to account for the varying non-rigid deformation behavior of different clothing during reconstruction. The non-linear optimization problems in both stages are solved with specially-tailored GPU accelerated Gauss-Newton solvers. In order to achieve a real-time performance of over 25 Hz, a pipelined solver architecture is designed that executes the first and the second stage on two GPUs in a rolling manner. The proposed approach captures high-quality models of humans and their clothing in real time from a single monocular camera. Intriguing examples of live applications in 3D video and virtual try-on are demonstrated. The proposed method, both qualitatively and quantitatively, outperforms related monocular online methods and comes close to offline performance capture approaches in terms of reconstruction density and accuracy.

In summary, the contributions of this chapter are:

- This chapter proposes the first real-time system for monocular human performance capture. In order to achieve real-time performance, not only specific algorithmic design choices are made, but also several new algorithmic ideas are proposed, e.g., the adaptive material-based regularization and the displacement warping to guarantee high-quality results under a tight real-time constraint.

- This chapter shows how to efficiently implement these design decisions by combining the compute power of two GPUs and the CPU in a pipelined architecture and how dense and sparse linear systems of equations can be efficiently optimized on the GPU.

- To evaluate the proposed approach on a wide range of data, high-quality results are shown on an extensive new dataset of more than 20 minutes of video footage captured in 11 scenarios, which contains different types of loose apparel and challenging motions.

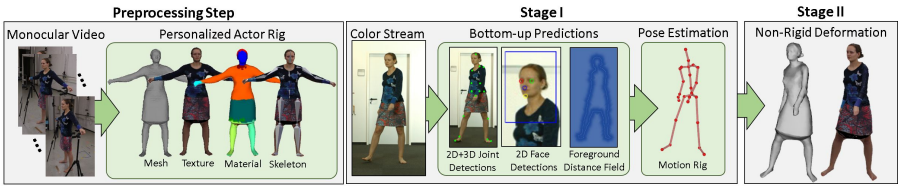


Figure 4.2: The proposed real-time performance capture approach reconstructs dense, space-time coherent deforming geometry of people in loose everyday clothing from just a single RGB stream. A skinned template is jointly fit to background-subtracted input video, 2D and 3D joint estimates, and sparse facial detections. Non-rigid 3D deformations of skin and even loose apparel are captured based on a novel real-time capable dense surface tracker.

4.2 OVERVIEW

The input to the proposed method is a single color video stream (Section 4.4). In addition, the approach requires a textured actor model, which is acquired in a preprocessing step (Section 4.3) from a monocular video sequence. From this input alone, this real-time human performance capture approach automatically estimates the articulated actor motion and the non-rigid deformation of skin and clothing in a coarse-to-fine manner from two subsequent stages per input frame. In the first stage, the articulated 3D pose of the underlying kinematic skeleton is estimated. To this end, an efficient way to fit the skeletal pose of the skinned template to 2D and 3D joint positions from a state-of-the-art CNN-based regressor to sparse detected face landmarks and to the foreground silhouette is proposed (Section 4.5). With this skeleton-deformed mesh and the warped non-rigid displacement of the previous frame as initialization, the second stage captures the surface deformation of the actor using a novel real-time template-to-image non-rigid registration approach (Section 4.6). The non-rigid registration is expressed as an optimization problem consisting of a silhouette alignment term, a photometric term, and several regularization terms; the formulation and combination of terms in the energy are geared towards high efficiency at high accuracy despite the monocular ambiguities. The different components of the proposed approach are illustrated in Figure 4.2. In order to achieve real-time performance, the underlying optimization problems are solved using dedicated data-parallel GPU optimizers (Section 4.7). In the following, all components are explained.

4.3 ACTOR MODEL ACQUISITION

Similar to many existing template-based performance capture methods (Allain et al., 2015; Cagniart et al., 2010; Gall et al., 2009; Vlasic et al., 2008; Xu et al., 2018), an actor model is reconstructed in a pre-

processing step. To this end, a set of M images $\mathcal{I}_{\text{rec}} = \{I_{\text{rec}_1}, \cdots, I_{\text{rec}_M}\}$ of the actor in a static neutral pose is taken from a video captured while walking around the person, which covers the entire body. For all the templates, around $M = 70$ images are used. With these images, a triangulated template mesh $\hat{\mathbf{V}} \in \mathbb{R}^{N\times 3}$ (N denotes the number of vertices in the mesh) and the associated texture map of the actor is generated using an image-based 3D reconstruction software[1]. The reconstructed geometry is downsampled to a resolution of approximately $N = 5000$ by using the Quadric Edge Collapse Decimation algorithm implemented in MeshLab[2]. The vertex colors of the template mesh $\mathbf{C} \in \mathbb{R}^{N\times 3}$ are transferred from the generated texture map. Then, skeleton joints and facial markers are manually placed on the template mesh resulting in a skeleton model. The template mesh is rigged to this skeleton model via dual quaternion skinning (Kavan et al., 2007) (see Chapter 3), where the skinning weights are automatically computed using Blender[3] (other auto-rigging tools would be feasible). This allows the proposed approach to deform the template mesh using the estimated skeletal pose parameters (Section 4.5). An important feature of this performance capture method is that material-dependent differences in the deformation behavior are explicitly modeled, e.g., of skin and apparel during tracking (see Section 4.6). To this end, a new multi-view method to segment the template into one of seven non-rigidity classes is proposed. First, the state-of-the-art human parsing method of Gong et al., 2017 is applied to each image in $\mathcal{I}_{\text{rec}}$ separately to obtain the corresponding semantic label images $\mathcal{L}_{\text{rec}} = \{L_{\text{rec}_1}, \cdots, L_{\text{rec}_M}\}$. The semantic labels $L \in \{1, \cdots, 20\}^N$ for all vertices $\mathbf{V}_i$ are computed based on their back-projection into all label images and a majority vote per vertex. The materials are binned into 7 non-rigidity classes, each having a different per-edge non-rigidity weight in the employed regularization term (Section 4.6). Those weights were empirically determined by visual observation of the deformation behavior under different weighting factors. The different classes and the corresponding non-rigidity weights are shown in Table 4.1. Very high weights are used for rigid body parts, e.g., the head, medium weights for the less rigid body parts, e.g., skin and tight clothing, and low weights for loose clothing. A high rigidity weight is used for any kind of hairstyle since, similar to all other human performance capture approaches, hair dynamics are not considered and thus not tracked. The per-vertex smoothness weights are mapped to per-edge non-rigidity weights $s_{i,j}$ by averaging the weights of vertex $\mathbf{V}_i$ and $\mathbf{V}_j$.

1 http://www.agisoft.com
2 http://www.meshlab.net/
3 https://www.blender.org/

Class ID	Weight	Part/Apparel Type
1	1.0	dress, coat, jumpsuit, skirt, background
2	2.0	upper clothes
3	2.5	pants
4	3.0	scarf
5	50.0	left leg, right leg, left arm, right arm, socks
6	100.0	hat, glove, left shoe, right shoe,
7	200.0	hair, face, sunglasses

Table 4.1: The employed non-rigidity weights $s_{i,j}$.

4.4 INPUT STREAM PROCESSING

After the actor model acquisition step, the proposed real-time performance capture approach works fully automatically, and the proposed method does not rely on careful initialization, e.g., it is sufficient to place the T-posed character model in the center of the frame. The input to the algorithm is a single color video stream from a static camera, e.g., a webcam. Thus, it is assumed that the camera and the world space are the same. The camera intrinsics are recovered using the Matlab calibration toolbox[4]. The skeletal pose estimation and non-rigid registration stages rely on the silhouette segmentation of the input video frames. To this end, the background subtraction method of Zivkovic and Heijden, 2006 is leveraged. It is assumed that the background is static, that its color is sufficiently different from the foreground, and that a few frames of the empty scene are recorded before performance capture commences. The distance transform images I_{DT} are efficiently computed from the foreground silhouettes, which are used in the skeletal pose estimation and non-rigid alignment step.

4.5 SKELETAL POSE ESTIMATION

The skeletal pose estimation is formulated as a non-linear optimization problem in the unknown skeleton parameters $\mathcal{S}^*$:

$$\mathcal{S}^* = \underset{\mathcal{S}}{\operatorname{argmin}}\, E_{\mathrm{pose}}(\mathcal{S}). \tag{4.1}$$

The set $\mathcal{S} = \{\boldsymbol{\theta}, \mathbf{R}, \mathbf{t}\}$ contains the joint angles $\boldsymbol{\theta} \in \mathbb{R}^{27}$ of the J joints of the skeletal model, and the global pose $\mathbf{R} \in \mathbf{SO}(3)$ and translation

4 http://www.vision.caltech.edu/bouguetj/calib_doc

$\mathbf{t} \in \mathbb{R}^3$ of the root. For pose estimation, an energy of the following general form is optimized:

$$\begin{aligned} E_{\text{pose}}(\mathcal{S}) =& E_{\text{2D}}(\mathcal{S}) + E_{\text{3D}}(\mathcal{S}) + E_{\text{silhouette}}(\mathcal{S}) + E_{\text{temporal}}(\mathcal{S}) \\ &+ E_{\text{anatomic}}(\mathcal{S}) \ . \end{aligned} \tag{4.2}$$

Here, E_{2D} and E_{3D} are alignment constraints based on the regressed 2D and 3D joint positions, respectively. In addition, $E_{\text{silhouette}}$ is a dense alignment term that fits the silhouette of the actor model to the detected silhouette in the input color images. Lastly, E_{temporal} and E_{anatomic} are temporal and anatomical regularization constraints that ensure that the speed of the motion and the joint angles stay in physically plausible ranges. To better handle fast motion, the skeleton parameters are initialized before optimization by extrapolating the poses of the last two frames in joint angle space based on an explicit Euler step. In the following, each energy term is explained in more detail.

4.5.1 *Sparse 2D and 3D Alignment Constraint*

For each input frame I, the 2D and 3D joint positions $\mathbf{P}_{\text{2D},i} \in \mathbb{R}^2$ and $\mathbf{P}_{\text{3D},i} \in \mathbb{R}^3$ of the J joints are estimated using the efficient deep skeleton joint regression network of the VNect algorithm (Mehta et al., 2017) trained with the original data of (Mehta et al., 2017). However, with these skeleton-only joint detections, it is not possible to determine the head's orientation. Therefore, the 2D joint predictions of (Mehta et al., 2017) are further augmented with a subset of the facial landmark detections of (Saragih et al., 2009), which includes the eyes, nose, and chin. The 2D detections $\mathbf{P}_{\text{2D},i} \in \mathbb{R}^2$ are incorporated based on the following re-projection constraint:

$$E_{\text{2D}}(\mathcal{S}) = \lambda_{\text{2D}} \sum_{i=1}^{J+4} \lambda_i \left\| \pi \left(p_{\mathbf{3D},i}(\boldsymbol{\theta}, \mathbf{R}, \mathbf{t}) \right) - \mathbf{P}_{\text{2D},i} \right\|^2 \ . \tag{4.3}$$

Here, $p_{\mathbf{3D},i}$ is the 3D position of the i-th joint/face marker of the used kinematic skeleton and $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is a full perspective projection that maps 3D space to the 2D image plane. Thus, this term enforces that all projected joint positions are close to their corresponding detections. λ_i are detection-based weights which are set to $\lambda_i = 0.326$ for the facial landmarks and $\lambda_i = 1.0$ for all other detections to avoid that the head error dominates all other body parts. To resolve the inherent depth ambiguities of the re-projection constraint, the following 3D-to-3D alignment term between model joints $p_{\mathbf{3D},i}(\boldsymbol{\theta}, \mathbf{R}, \mathbf{t})$ and 3D detections $\mathbf{P}_{\text{3D},i}$ is employed:

$$E_{\text{3D}}(\mathcal{S}) = \lambda_{\text{3D}} \sum_{i=1}^{J} \left\| p_{\mathbf{3D},i}(\boldsymbol{\theta}, \mathbf{R}, \mathbf{t}) - \left(\mathbf{P}_{\text{3D},i} + \mathbf{t}' \right) \right\|^2 \ . \tag{4.4}$$

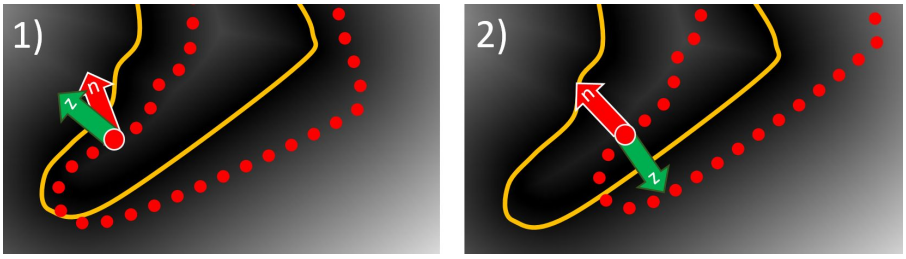


Figure 4.3: The two cases in the silhouette alignment constraint. Target silhouette (yellow), model silhouette (red), the negative gradient of the distance field $\mathbf{z}$ (green arrow), and the projected 2D normal $\mathbf{n}$ of the boundary vertex (red arrow).

Here, $\mathbf{t}' \in \mathbb{R}^3$ is an auxiliary variable that transforms the regressed 3D joint positions $\mathbf{P}_{3D,i}$ from the root centered local coordinate system to the global coordinate system. Note that the regressed 3D joint positions $\mathbf{P}_{3D,i}$ are in normalized space. Therefore, the regressed skeleton is rescaled according to the bone lengths of the parameterized skeleton model.

4.5.2 *Dense Silhouette Alignment Constraint*

A dense alignment between the boundary of the skinned actor model and the detected silhouette in the input image is also enforced. In contrast to the approach of Xu et al., 2018, which requires closest point computations, a distance transform-based constraint is employed for efficiency reasons. Once per frame, a set of contour vertices $\mathcal{B}$ is extracted from the current deformed version of the actor model. Afterwards, all contour vertices are forced to align well to the interface between the detected foreground and background:

$$E_{\text{silhouette}}(\mathcal{S}) = \lambda_{\text{silhouette}} \sum_{i \in \mathcal{B}} b_i \cdot \left[I_{\text{DT}}\big(\pi(\mathbf{V}_i(\boldsymbol{\theta}, \mathbf{R}, \mathbf{t}))\big) \right]^2 . \tag{4.5}$$

Here, $\mathbf{V}_i$ is the i-th boundary vertex of the skinned actor model, and the image I_{DT} stores the Euclidean distance transform with respect to the detected silhouette in the input image. The $b_i \in \{-1, +1\}$ are directional weights that guide the optimization to follow the right direction in the distance field. In the minimization of the term in Equation 4.5, silhouette model points are pushed in the negative direction of the distance transform image gradient $\mathbf{z} = -\nabla_{xy} I_{DT} \in \mathbb{R}^2$. By definition, $\mathbf{z}$ points in the direction of the nearest *image silhouette* (IS) contour. If model points fall outside of the IS, they will be dragged towards the nearest IS contour as desired. However, when model points fall inside the IS, there are two possibilities: 1) the model point normal $\mathbf{n}$ follows roughly the same direction as $\mathbf{z}$ or 2) it does not. In case 1) the normal at the nearest IS point matches the direction of the model

point normal. This indicates that $\mathbf{z}$ is a good direction to follow. In case 2) however, the normal at the nearest IS point follows the opposite direction, indicating that $\mathbf{z}$ is pointing towards the wrong IS contour, see Figure 4.3. Therefore, in case 2) the opposite direction $\mathbf{p} = -\mathbf{z}$ is chosen by setting $b_i = -1$. This is preferable over just following $\mathbf{n}$, since $\mathbf{n}$ is not necessarily pointing away from the wrong IS contour. Mathematically, case 2) is considered when $\mathbf{n}^T\mathbf{z} < 0$. For all the other cases, the direction of $\mathbf{z}$ is chosen by setting $b_i = +1$.

4.5.3 *Temporal Stabilization*

To mitigate temporal noise, a temporal stabilization constraint is used, which penalizes the change in joint position between the current and previous frame:

$$E_{\text{temporal}}(\mathcal{S}) = \lambda_{\text{temporal}} \sum_{i=1}^{J} \lambda_i \left\| p_{\mathbf{3D},i}(\boldsymbol{\theta}, \mathbf{R}, \mathbf{t}) - p^{t-1}_{\mathbf{3D},i}(\boldsymbol{\theta}, \mathbf{R}, \mathbf{t}) \right\|^2 . \quad (4.6)$$

Here, the λ_i are joint-based temporal smoothness weights which are set to $\lambda_i = 2.5$ for joints on the torso and the head, $\lambda_i = 2.0$ for shoulders, $\lambda_i = 1.5$ for knees and elbows, and $\lambda_i = 1.0$ for the hands and feet.

4.5.4 *Joint Angle Limits*

The joints of the human skeleton have physical limits. This prior knowledge is integrated into the pose estimation objective based on a soft-constraint on $\boldsymbol{\theta} \in \mathbb{R}^{27}$. To this end, all degrees of freedom are forced to stay within their anatomical limits $\boldsymbol{\theta}_{\text{min}} \in \mathbb{R}^{27}$ and $\boldsymbol{\theta}_{\text{max}} \in \mathbb{R}^{27}$:

$$E_{\text{anatomic}}(\mathcal{S}) = \lambda_{\text{anatomic}} \sum_{i=1}^{27} \Psi(\boldsymbol{\theta}_i) \ .$$

Here, $\Psi(x)$ is a quadratic barrier function that penalizes if a degree of freedom exceeds its limits:

$$\Psi(x) = \begin{cases} (x - \boldsymbol{\theta}_{\text{max},i})^2, & \text{if } x > \boldsymbol{\theta}_{\text{max},i} \\ (\boldsymbol{\theta}_{\text{min},i} - x)^2, & \text{if } x < \boldsymbol{\theta}_{\text{min},i} \\ 0 & , \text{ otherwise } . \end{cases}$$

This term prevents implausible human pose estimates.

4.6 NON-RIGID SURFACE REGISTRATION

The pose estimation step cannot capture realistic non-rigid deformations of skin and clothing that are not explained through skinning. The model, therefore, does not yet align with the image well everywhere,

in particular around the cloth and in some skin regions. Hence, starting from the pose estimation result, the following non-rigid surface tracking energy is minimized:

$$E_{\text{non-rigid}}(\mathbf{V}) = E_{\text{data}}(\mathbf{V}) + E_{\text{reg}}(\mathbf{V}) \ . \tag{4.7}$$

The energy consists of several data terms E_{data} and regularization constraints E_{reg}, which is explained in the following. The proposed data terms are a combination of a dense photometric alignment term E_{photo} and a dense silhouette alignment term $E_{\text{silhouette}}$:

$$E_{\text{data}}(\mathbf{V}) = E_{\text{photo}}(\mathbf{V}) + E_{\text{silhouette}}(\mathbf{V}) \ . \tag{4.8}$$

4.6.1 *Dense Photometric Alignment*

The photometric alignment term measures the re-projection error densely:

$$E_{\text{photo}}(\mathbf{V}) = \sum_{i \in \mathcal{V}} w_{\text{photo}} \left\| \sigma_c \left(I_{\text{Gauss}} \left(\pi \left(\mathbf{V}_i \right) \right) - \mathbf{C}_i \right) \right\|^2 , \tag{4.9}$$

where $\mathbf{C}_i$ is the color of vertex $\mathbf{V}_i$ in the template model and $\sigma_c(\cdot)$ is a robust kernel that prunes wrong correspondences according to color similarity by setting residuals that are above a certain threshold to zero. More specifically, every visible vertex $\mathbf{V}_i \in \mathcal{V}$ is projected onto the screen space based on the full perspective camera model π. The visibility is obtained from the skinned mesh after the pose estimation step using depth buffering. In order to speed up convergence, the photometric term is computed based on a 3-level pyramid of the input image where one Gauss-Newton iteration is performed on each level. The projected positions are used to sample a Gaussian blurred version I_{Gauss} of the input image I at the current time step for more stable and longer range gradients. The Gaussian kernel sizes for the 3 levels are 15, 9, and 3, respectively.

4.6.2 *Dense Silhouette Alignment*

In addition to dense photometric alignment, an alignment of the projected 3D model boundary with the detected silhouette in the input image is also enforced:

$$E_{\text{silhouette}}(\mathbf{V}) = w_{\text{silhouette}} \sum_{i \in \mathcal{B}} b_i \cdot \left[I_{\text{DT}}\big(\pi \left(\mathbf{V}_i \right) \big) \right]^2 . \tag{4.10}$$

After Stage I, the model boundary $\mathcal{B}$ is first updated, and all vertices $\mathbf{V}_i \in \mathcal{B}$ are considered. These boundary vertices are encouraged to match the zero iso-line of the distance transform image I_{DT} and thus be aligned with the detected input silhouette. The b_i are computed similar

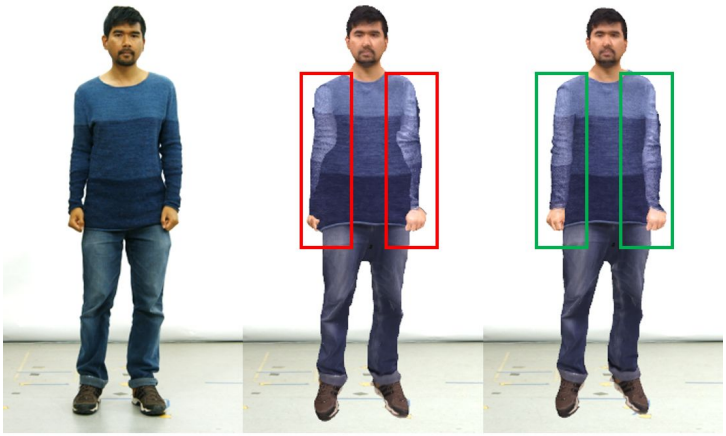


Figure 4.4: Left: Input image. Middle: Textured reconstruction without using the body part mask. One can clearly see the artifacts since multiple model boundaries wrongly explain the silhouette of the arms. Right: Using the body part mask in the distance transform image, the foreground silhouette is correctly explained.

to the pose optimization step (see Section 4.5). Due to the non-rigid deformation that cannot be recovered by the pose estimation stage, in some cases, the projection of the mesh from Stage I has a gap between body parts such as arms and torso, while in the input image, the gaps do not exist. To prevent image silhouettes being wrongly explained by multiple model boundaries, the posed model $\mathbf{V}^S$ is projected into the current frame, and a body part mask is computed — derived from the skinning weights. The extent of each body part is increased by a dilation (maximum of 10 pixels, the torso is preferred over the other parts) to obtain a conservative region boundary that closes the above-mentioned gaps. If a vertex $\mathbf{V}_i$ moves onto a region with a differing semantic label, its silhouette term is disabled by setting $b_i = 0$. This drastically improves the reconstruction quality (see Figure 4.4).

This high-dimensional monocular non-rigid registration problem with only the data terms is ill-posed. Therefore, regularization constraints are used:

$$E_{\text{reg}}(\mathbf{V}) = E_{\text{smooth}}(\mathbf{V}) + E_{\text{edge}}(\mathbf{V}) + E_{\text{velocity}}(\mathbf{V}) + E_{\text{acceleration}}(\mathbf{V}) \ . \tag{4.11}$$

Here, E_{smooth} and E_{edge} are spatial smoothness priors on the mesh geometry, and E_{velocity} and $E_{\text{acceleration}}$ are temporal priors. In the following, more details are provided.

4.6.3 *Spatial Smoothness*

The first prior on the mesh geometry is a spatial smoothness term with respect to the pose estimation result:

$$E_{\text{smooth}}(\mathbf{V}) = w_{\text{smooth}} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{s_{ij}}{|\mathcal{N}_i|} \left\| (\mathbf{V}_i - \mathbf{V}_j) - (\mathbf{V}_i^{\text{S}} - \mathbf{V}_j^{\text{S}}) \right\|^2 . \tag{4.12}$$

Here, the $\mathbf{V}_i$ are the unknown optimal vertex positions, and the $\mathbf{V}_i^{\text{S}}$ are the vertex positions after skinning using the current pose estimation result of Stage I. s_{ij} are the semantic label-based per-edge smoothness weights (see Section 4.3) that model material dependent non-rigidity. The energy term enforces that every edge in the deformed model is similar to the undeformed model regarding its length and orientation. In addition to this surface smoothness term, locally isometric deformations are also enforced:

$$E_{\text{edge}}(\mathbf{V}) = w_{\text{edge}} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \frac{s_{ij}}{|\mathcal{N}_i|} \left(\left\| \mathbf{V}_i - \mathbf{V}_j \right\| - \left\| \hat{\mathbf{V}}_i - \hat{\mathbf{V}}_j \right\| \right)^2 , \tag{4.13}$$

where $\hat{\mathbf{V}}$ denotes the vertex position in the template's rest pose. It is enforced that the edge length does not change much between the rest pose $\hat{\mathbf{V}}_i$ and the optimal unknown pose $\mathbf{V}_i$. While this is similar to the first term, it allows the approach to penalize stretching independently of shearing.

4.6.4 *Temporal Smoothness*

Temporal priors are also employed to favor temporally coherent non-rigid deformations. Similar to temporal smoothness in skeletal pose estimation, the first term

$$E_{\text{velocity}}(\mathbf{V}) = w_{\text{velocity}} \sum_{i=1}^{N} \left\| \mathbf{V}_i - \mathbf{V}_i^{t-1} \right\|^2 , \tag{4.14}$$

encourages small velocity, and the second term

$$E_{\text{acceleration}}(\mathbf{V}) = w_{\text{acceleration}} \sum_{i=1}^{N} \left\| \mathbf{V}_i - 2\mathbf{V}_i^{t-1} + \mathbf{V}_i^{t-2} \right\|^2 , \tag{4.15}$$

encourages small acceleration between adjacent frames.

4.6.5 *Displacement Warping*

The non-rigid displacements $\mathbf{d}_i^{t-1} = \mathbf{V}_i^{t-1} - \mathbf{V}_i^{\text{S},t-1} \in \mathbb{R}^3$ that are added to each vertex i after skinning are usually similar from frame $t-1$ to frame t. $\mathbf{d}_i^{t-1}$ is warped back to the rest pose by applying Dual

Quaternion skinning with the inverse rotation quaternions given by the pose at time $t-1$. They are referred to as $\hat{\mathbf{d}}_i^{t-1}$. For the next frame t, $\hat{\mathbf{d}}_i^{t-1}$ is transformed according to the pose at time t resulting in a skinned displacement $\mathbf{d}_i^{S,t}$. Then, the non-rigid stage is initialized with $\mathbf{V}_i^t = \mathbf{V}_i^{S,t} + \mathbf{d}_i^{S,t}$. This jump-starts the non-rigid alignment step and leads to improved tracking quality. Similarly, $\mathbf{d}_i^{S,t}$ is added to the skinned actor model for a more accurate dense silhouette alignment during the skeletal pose estimation stage.

4.6.6 *Vertex Snapping*

After the non-rigid stage, the boundary vertices are already very close to the image silhouette. Therefore, they can be robustly snapped to the closest silhouette point by walking on the distance transform along the negative gradient direction until the zero crossing is reached. Vertex snapping allows the algorithm to reduce the number of iteration steps since if the solution is already close to the optimum, the updates of the solver become smaller, as is true for most optimization problems. Therefore, if the mesh is already close to the silhouette, it is 'snapped' to the silhouette in a single step instead of requiring multiple iterations of Gauss-Newton. To obtain continuous results, non-boundary vertices are smoothly adjusted based on a Laplacian warp in a local neighborhood around the mesh contour.

4.7 DATA-PARALLEL GPU OPTIMIZATION

The described pose estimation and non-rigid registration problems are non-linear optimizations based on an objective E with respect to unknowns $\mathcal{X}$, i.e., the parameters of the kinematic model $\mathcal{S}$ for pose estimation and the vertex positions $\mathbf{V}$ for non-rigid surface deformation. The optimal parameters $\mathcal{X}^*$ are found via energy minimization:

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} E(\mathcal{X}) \ . \tag{4.16}$$

In both capture stages, i.e., pose estimation (see Section 4.5) and non-rigid surface tracking (see Section 4.6), the objective E can be expressed as a sum of squares:

$$E(\mathcal{X}) = \sum_i \left[\mathbf{F}_i(\mathcal{X})\right]^2 = \left|\left|\mathbf{F}(\mathcal{X})\right|\right|_2^2 \ . \tag{4.17}$$

Here, $\mathbf{F}$ is the error vector resulting from stacking all residual terms. This optimization is solved at real-time rates using a data-parallel iterative Gauss-Newton solver that minimizes the total error by linearizing $\mathbf{F}$ and taking local steps $\mathcal{X}_k = \mathcal{X}_{k-1} + \delta_k^*$ obtained by the solution of a sequence of linear sub-problems (normal equations):

$$\mathbf{J}^T(\mathcal{X}_{k-1})\mathbf{J}(\mathcal{X}_{k-1}) \cdot \delta_k^* = -\mathbf{J}^T(\mathcal{X}_{k-1})\mathbf{F}(\mathcal{X}_{k-1}) \ . \tag{4.18}$$

Here, $\mathbf{J}$ is the Jacobian of $\mathbf{F}$. Depending on the problems (pose estimation or non-rigid registration), the linear systems have a quite different structure in terms of dimensionality and sparsity. Thus, tailored parallelization strategies for each of the problems are used. Since Gauss-Newton is used instead of Levenberg-Marquardt, the residual has not to be computed during the iterations, thus leading to faster runtimes, and in consequence, more iterations are possible within the tight real-time constraint.

4.7.1 *Pose Estimation*

The normal equations of the pose optimization problem are small but dense, i.e., the corresponding system matrix is small, rectangular, and dense. Handling each non-linear Gauss-Newton step efficiently requires a specifically tailored parallelization and optimization strategy. First, at the beginning of each Gauss-Newton step, the system matrix $\mathbf{J}^T\mathbf{J}$ and the right-hand side $-\mathbf{J}^T\mathbf{F}$ are computed in global memory on the GPU. Afterwards, the small system of size 36×36 (36 = 3+3+27+3, 3 DoFs for $\mathbf{R}$, 3 for $\mathbf{t}$, 27 for θ, and 3 for $\mathbf{t}'$) is shipped to the CPU and it is solved using QR decomposition. The strategy of splitting the computation to CPU and GPU is in spirit similar to (Tagliasacchi et al., 2015). To compute $\mathbf{J}^T\mathbf{J}$ on the GPU, $\mathbf{J}$ is first computed fully in parallel, and it is then stored in device memory based on a kernel that launches one thread per matrix entry. A similar operation is performed for $\mathbf{F}$. $\mathbf{J}^T\mathbf{J}$ is then computed based on a data-parallel version of a matrix-matrix multiplication that exploits shared memory for high performance. The same kernel also directly computes $\mathbf{J}^T\mathbf{F}$. Several thread blocks are launched per element of the output matrix/vector, which cooperate in computing the required dot products, e.g., between the i-th and j-th column of $\mathbf{J}$ or the i-th column of $\mathbf{J}$ and $\mathbf{F}$. To this end, each thread block computes a small subpart of the dot product based on a shared memory reduction. The per-block results are summed up based on global memory atomics. In total, 6 Gauss-Newton steps are performed, which turned out to be a good trade-off between accuracy and speed.

4.7.2 *Non-rigid Surface Registration*

The non-rigid optimization problem that results from the energy $E_{\text{non-rigid}}$ has a substantially different structure. It leads to a large sparse system of normal equations, i.e., the corresponding system matrix is sparse and has a low number of non-zeros per row. Similar to Zollhöfer et al., 2014 and Innmann et al., 2016, during GPU-based data-parallel Preconditioned Conjugate Gradient (PCG), the solver is parallelized over the rows (unknowns) of the system matrix $\mathbf{J}^T\mathbf{J}$ using one thread per block row (x-,y-, and z-entry of a vertex). Each

thread collects and handles all non-zeros in the corresponding row. The diagonal of $\frac{1}{\mathbf{J}^T\mathbf{J}}$ is used as a preconditioner. Three Gauss-Newton steps are performed, and the linear system is solved based on 4 PCG iterations, which turned out to be a good trade-off between accuracy and speed.

4.7.3 *Pipelined Implementation*

To achieve real-time performance, a data-parallel implementation of the entire performance capture algorithm is used in combination with a pipeline strategy tailored for the dedicated problem. To this end, the proposed approach is running three threads on a PC with two GPUs. Thread 1 uses only the CPU, which is responsible for data preprocessing. Thread 2 computes the CNN-based human pose detection on the first graphics card, thread 3 solves the pose optimization problem and estimates the non-rigid deformation on the second graphics card. The proposed distributed computation strategy induces a 2 frame delay, but for most applications, it is barely noticeable.

4.8 EVALUATION

For all the tests, an Intel Core i7 is employed with two Geforce GTX 1080Ti graphics cards. The algorithm runs at around 25 FPS, which fulfills the performance requirement of many real-time applications. In all the experiments, the same set of parameters are used which are empirically determined: $\lambda_{2D} = 460$, $\lambda_{3D} = 28$, $\lambda_{\text{silhouette}} = 200$, $\lambda_{\text{temporal}} = 1.5$, $\lambda_{\text{anatomic}} = 10^6$, $w_{\text{photo}} = 10000$, $w_{\text{silhouette}} = 600$, $w_{\text{smooth}} = 10.0$, $w_{\text{edge}} = 30.0$, $w_{\text{velocity}} = 0.25$ and $w_{\text{acceleration}} = 0.1$. In the following, a new dataset is introduced. Then, the proposed approach is qualitatively and quantitatively evaluated on several challenging sequences, and it is also compared to related methods. After that, an ablation evaluation is performed to study the importance of the different components of the proposed approach. Finally, several live applications are demonstrated. A smoothing step with a filter of window size 3 (stencil: $[0.15, 0.7, 0.15]$) is applied to the trajectories of the vertex coordinates as a post-process for all video results except in the live setup.

4.8.1 *Dataset*

In order to qualitatively evaluate the proposed method on a wide range of settings, several challenging motion sequences are recorded. These contain large variations in non-rigid clothing deformations, e.g., skirts and hooded sweaters, and fast motions like dancing and jumping jacks. In total, over 20 minutes of video footage are captured that is split

Figure 4.5: Qualitative results. Several live monocular performance capture results of entire humans are shown in their loose everyday clothing. (a) shows the template models. (b) shows input images to the proposed method, while (c) shows that the corresponding results precisely overlay the person in the input images. The results can be used to render realistic images (d) or free-viewpoint video (e).

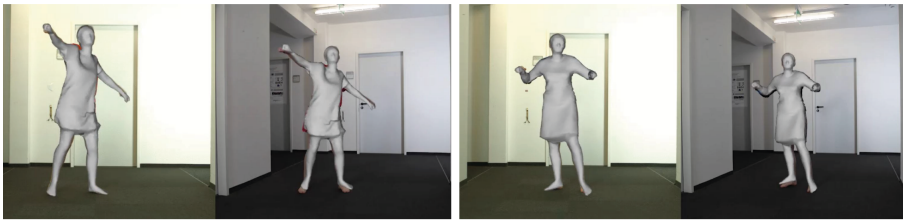


Figure 4.6: The accurate overlay on the input view with the reconstructed mesh shows the high quality of the reconstruction. Also, note that the reconstruction results match the images captured from a laterally displaced reference view, which is not used for tracking.

into 11 sequences with different sets of apparel, each worn by one of seven subjects. All sequences were recorded with a Blackmagic video camera (30fps, 540×960 resolution). The dataset provides semantically segmented, rigged, and textured templates, calibrated camera parameters, and an empty background image for all sequences. In addition, it also contains the silhouettes from background subtraction, the motion estimates, and the non-rigidly deformed meshes. For eight sequences, the subject was captured from a reference view, which will also be made available to evaluate the tracking quality. Figure 4.5 shows some of the templates and example frames of the captured sequences. The full dataset will be made publicly available.

4.8.2 *Evaluation Setup*

In total, the proposed approach is evaluated on the new dataset and five existing video sequences of people in different sets of apparel. In addition, the method is tested with 4 subjects in a live setup (see Figure 4.1) with a low-cost webcam. The method takes frames at 540×960 resolution as input. To better evaluate the non-rigid surface registration method, challenging loose clothing is used in these sequences, including skirts, dresses, hooded sweatshirts, and baggy pants. The sequences show a wide range of difficult motions (slow to fast, self-occlusions) for monocular capture. Additionally, the proposed approach is compared to the state-of-the-art monocular performance capture method of Xu et al., 2018 on two of their sequences and on one of the new captured sequences.

4.8.3 *Qualitative Evaluation*

In Figure 4.5, several frames from live performance capture results are shown. The results precisely overlay the person in the input images. Note that body pose, head orientation, and non-rigid deformation of loose clothing are accurately captured. Both the side-by-side compar-

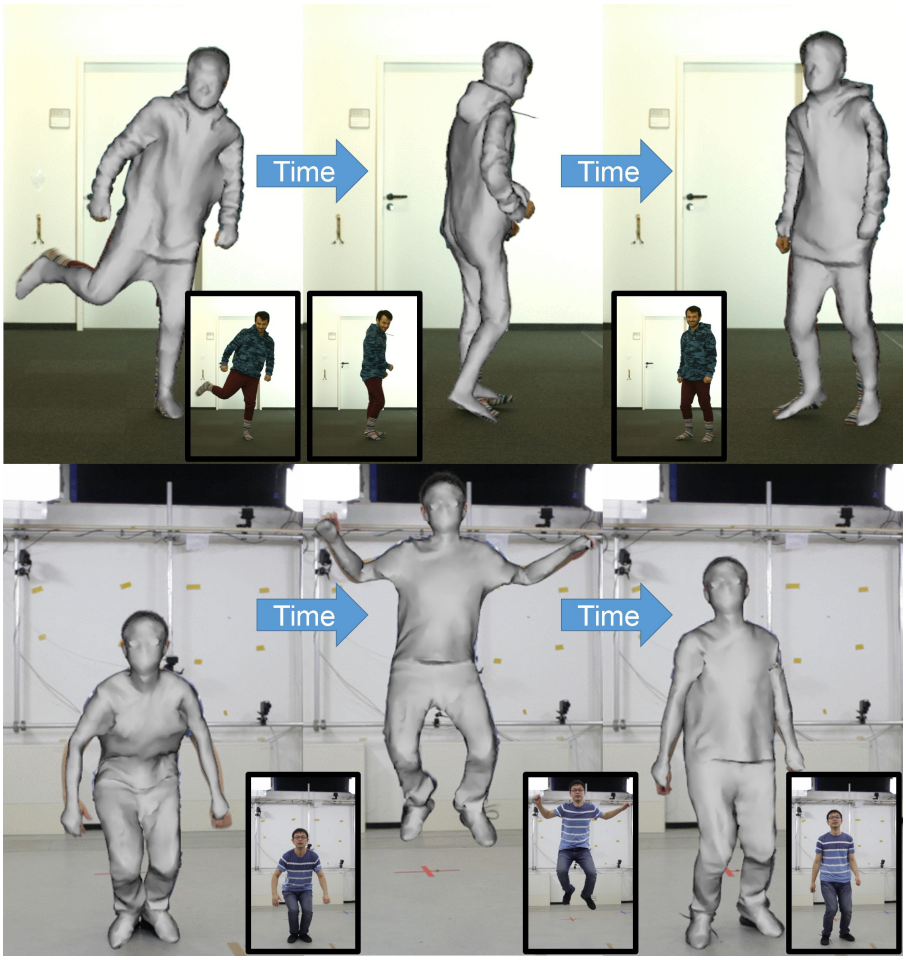


Figure 4.7: The proposed real-time approach even tracks challenging and fast motions, such as jumping and a fast 360° rotation with high accuracy. The reconstructions overlay the input image well.

ison to RGB input and the accurate overlay with the reconstructed mesh show the high quality of the reconstruction. Also, note that the reconstruction results match the images captured from a laterally displaced reference view, which is not used for tracking (see Fig. 4.6). This further evidences the fidelity of the 3D performance capture results, also in-depth, which shows that the formulation effectively meets the non-trivial underconstrained monocular reconstruction challenge. To evaluate the robustness of the proposed method, many fast and challenging motions are included in the test set. As shown in Figure 4.7, even the fast 360° rotation (see the first row) and the jumping motion (see the second row) are successfully tracked. This illustrates the robustness of the algorithm and its efficient and effective combined consideration of sparse and dense image cues, as well as learning-based and model-based capture, which in this combination were not used in prior work, let alone in real time.

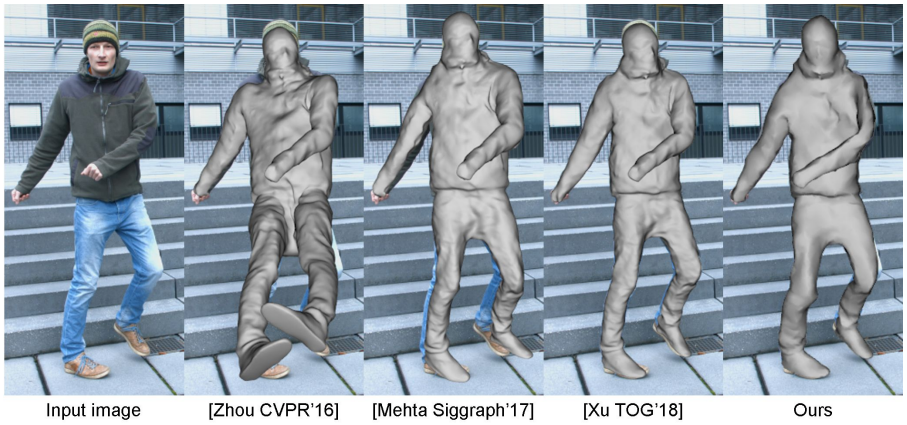


Figure 4.8: Qualitative comparison to related monocular methods. The results of the proposed approach overlay much better with the input than the skeleton-only results of Zhou et al., 2016 and Mehta et al., 2017. The shown results come close in quality to the off-line approach of Xu et al., 2018.

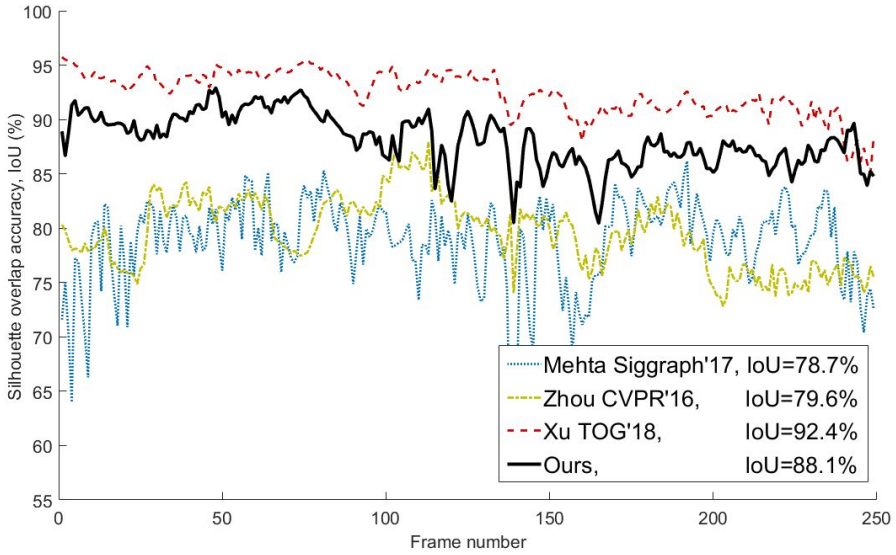


Figure 4.9: Quantitative comparison to related monocular methods. In terms of the silhouette overlap accuracy (Intersection over Union, IoU), the proposed method achieves better results and outperforms (Zhou et al., 2016) and (Mehta et al., 2017) by 8.5% and 9.4%, respectively. On average, the results are only 4.3% worse than the off-line approach of Xu et al., 2018, but the proposed approach is orders of magnitude faster.

4.8.4 *Comparison to Related Monocular Methods*

In Figure 4.8, a comparison to 3 related state-of-the-art methods is provided: The fundamentally off-line, monocular dense (surface-based) performance capture method of Xu et al., 2018, called MonoPerfCap, and two current monocular methods for 3D skeleton-only reconstruction, the 2D-to-3D lifting method of Zhou et al., 2016 and the real-time VNect algorithm (Mehta et al., 2017). For the latter two, the skinned rendering of the template using their skeleton pose is shown. The test sequence is provided by Xu et al., 2018 with manually labeled ground truth silhouettes. The proposed method's results overlay much better with the input than the skeleton-only results of Zhou et al., 2016 and Mehta et al., 2017, confirming the much better reconstructions. Also, a quantitative comparison on this sequence in terms of the silhouette overlap accuracy (Intersection over Union, IoU), Figure 4.9, shows that the proposed method achieves clearly better results and outperforms (Zhou et al., 2016) and (Mehta et al., 2017) by 8.5% and 9.4%, respectively. Using the same metric, the IoU is only 4.3% smaller than Xu et al., 2018, which is mainly caused by the fact that their foreground segmentation is more accurate due to their more advanced but offline foreground segmentation strategy (see Figure 4.11). However, please note that the proposed method is overall orders of magnitude faster than their algorithm, which takes over 1 minute per frame, and the reconstructions are still robust to the noisy foreground segmentation. To compare against MonoPerfCap more thoroughly, it is also evaluated on one of the sequences of the new dataset (see Section 4.8.1), which shows more challenging non-rigid dress deformations in combination with fast motions (see bottom rows of Figure 4.10). On this sequence, the accuracy of the foreground estimation is roughly the same, leading to the fact that the proposed approach achieves an IoU of 86.86% (averaged over 500 frames), which is almost identical to the one of Xu et al., 2018 (86.89%). As shown in Figure 4.10, comparable reconstruction quality, and overlay is achieved while being orders of magnitude faster. MonoPerfCap's window-based optimizer achieves slightly better boundary alignment and more stable tracking for some difficult, convolved, and self-occluded poses but is much slower. The reconstruction of the head and feet is consistently better than (Xu et al., 2018) due to the additional facial landmark alignment term and the better pose detector that are employed.

4.8.5 *Surface Reconstruction Accuracy*

To evaluate the surface reconstruction error, also relative to multi-view methods, the *Pablo* sequence from the state-of-the-art multi-view template-based performance capture method of Robertini et al., 2016 (they also provide the template) is used. As shown in Figure 4.13, the

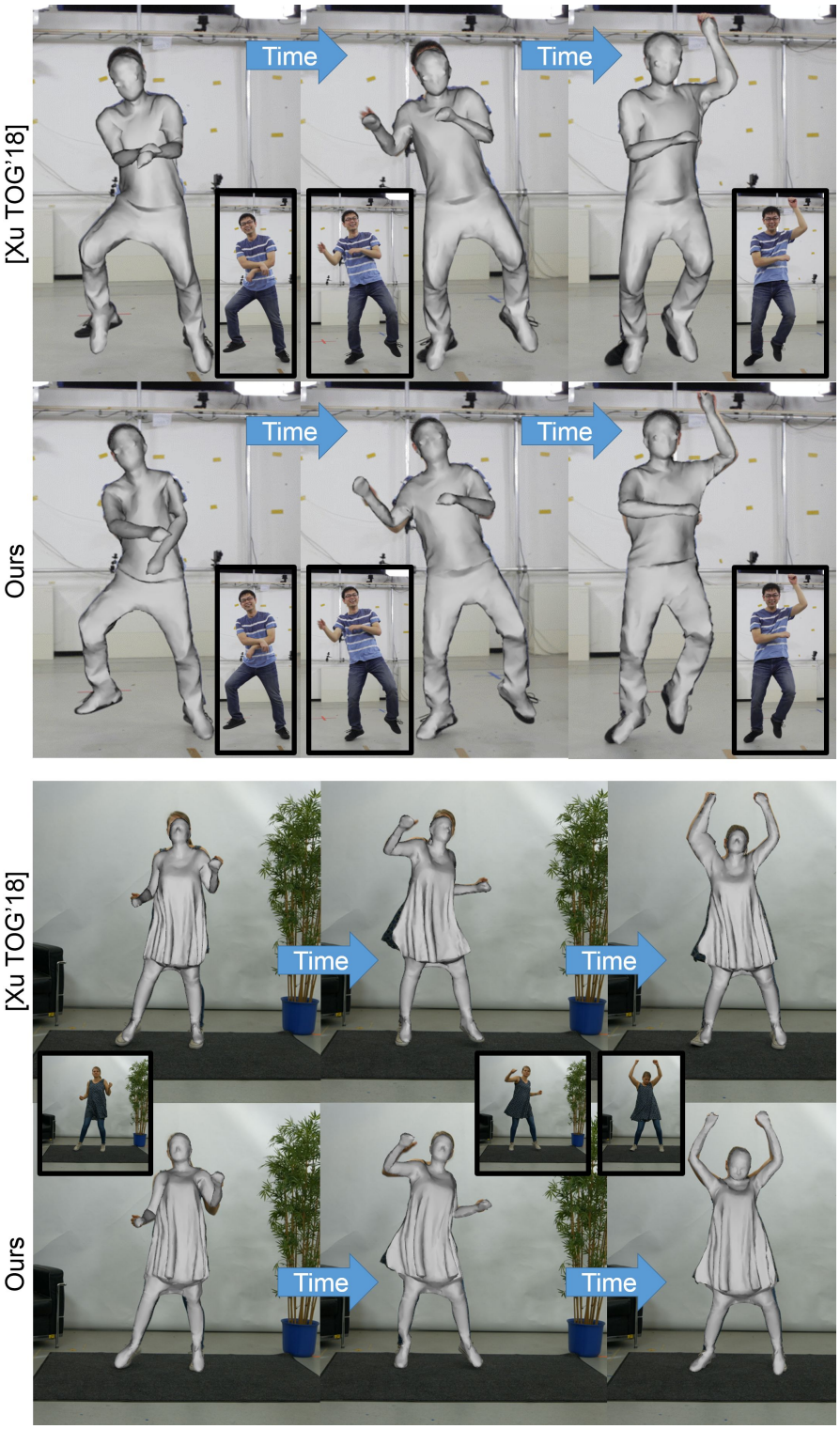


Figure 4.10: Qualitative comparison to MonoPerfCap (Xu et al., 2018). Comparable reconstruction quality and overlay are achieved while being orders of magnitude faster.

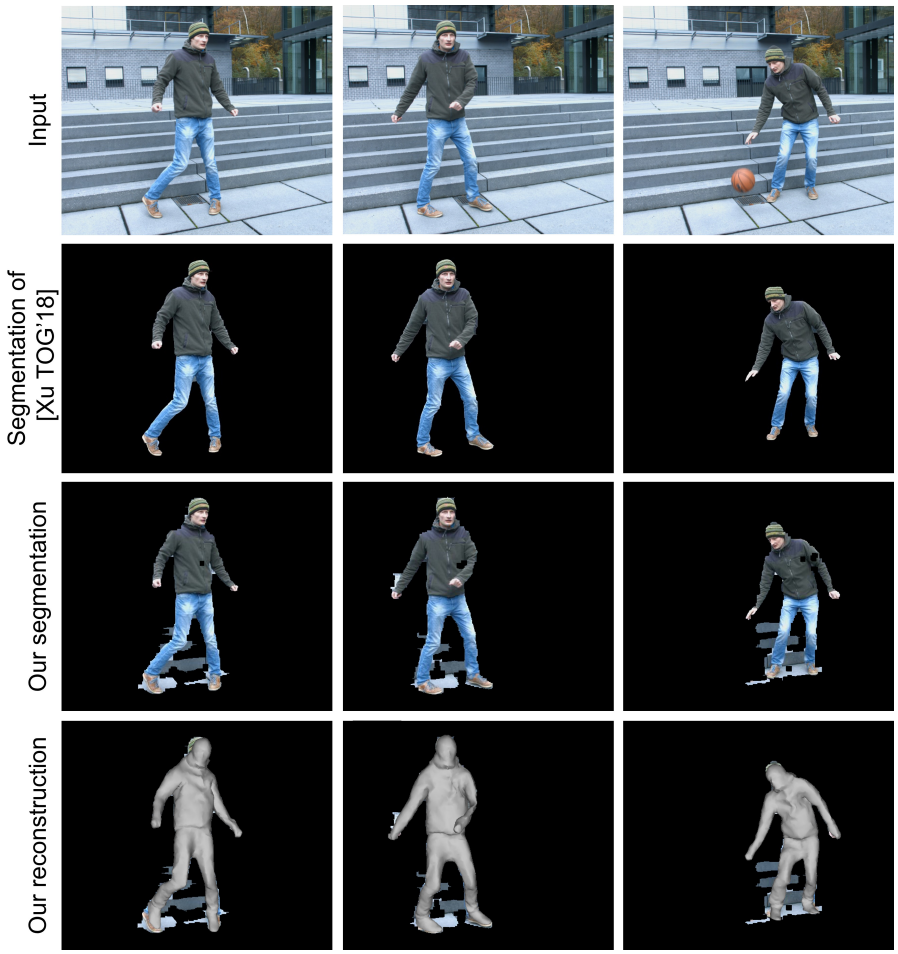


Figure 4.11: Comparison of the foreground segmentation of Xu et al., 2018 and the proposed method. Note that the silhouette estimates are less accurate than the ones of Xu et al., 2018. Nevertheless, the reconstruction results are robust to the noisy foreground estimates and look plausible.

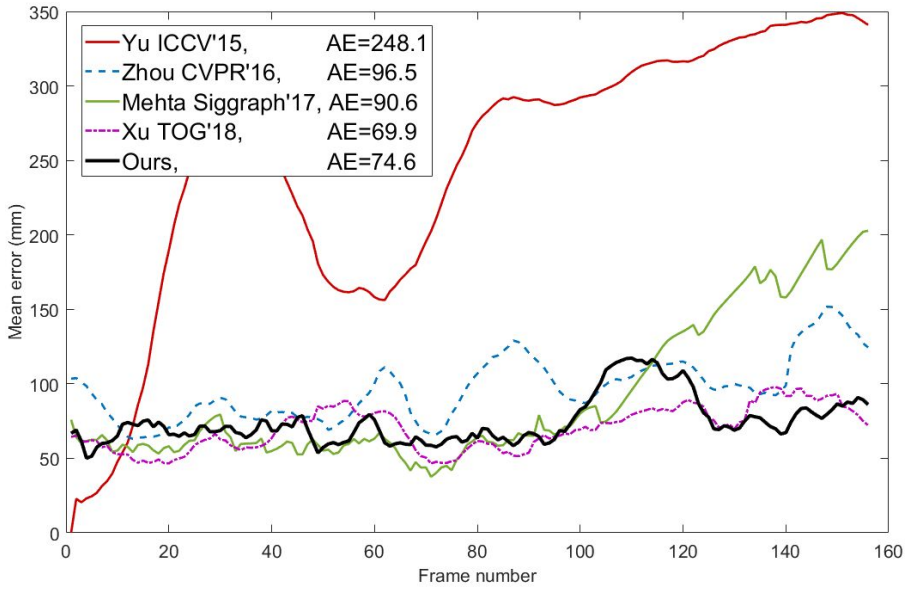


Figure 4.12: Quantitative comparison of the surface reconstruction accuracy on the *Pablo* sequence. The proposed real-time monocular approach comes very close in quality to the results of the monocular *offline* method of Xu et al., 2018. It clearly outperforms the monocular non-rigid capture method of Yu et al., 2015 and the rigged skeleton-only results of the 3D pose estimation methods of Zhou et al., 2016 and Mehta et al., 2017.

proposed real-time monocular method comes very close in quality to the results of the fundamentally off-line *multi-view* approach of Robertini et al., 2016 and the monocular *off-line* method of Xu et al., 2018. In addition, it clearly outperforms the monocular non-rigid capture method of Yu et al., 2015 and the rigged skeleton-only results of the 3D pose estimation methods of Zhou et al., 2016 and Mehta et al., 2017 (latter two as described in the previous paragraph). This is further evidenced by the quantitative evaluation of per-vertex position errors (see Figure 4.12). The reconstruction results of Robertini et al., 2016 is used as a reference, and the per-vertex Euclidean surface error is shown. Similar to (Xu et al., 2018), the reconstruction of all methods are aligned to the reference meshes with a translation to eliminate the global depth offset. The method of Xu et al., 2018 achieves slightly better results in terms of surface reconstruction accuracy. Similar to the previous experiment (see Figure 4.11), it can be observed that the foreground estimates are slightly worse than the ones of Xu et al., 2018 which caused the lower accuracy.

4.8.6 *Skeletal Pose Estimation Accuracy*

The proposed approach is also compared against VNect (Mehta et al., 2017), (Zhou et al., 2016) and MonoPerfCap (Xu et al., 2018) in terms of joint position accuracy on the *Pablo* sequence. As a reference, the

Figure 4.13: Qualitative comparisons of the surface reconstruction accuracy on the *Pablo* sequence. The proposed real-time monocular approach comes very close in quality to the results of the fundamentally off-line *multi-view* approach of Robertini et al., 2016 and the monocular *off-line* method of Xu et al., 2018. It clearly outperforms the monocular non-rigid capture method of Yu et al., 2015 and the rigged skeleton-only results of the 3D pose estimation methods of Zhou et al., 2016 and Mehta et al., 2017.

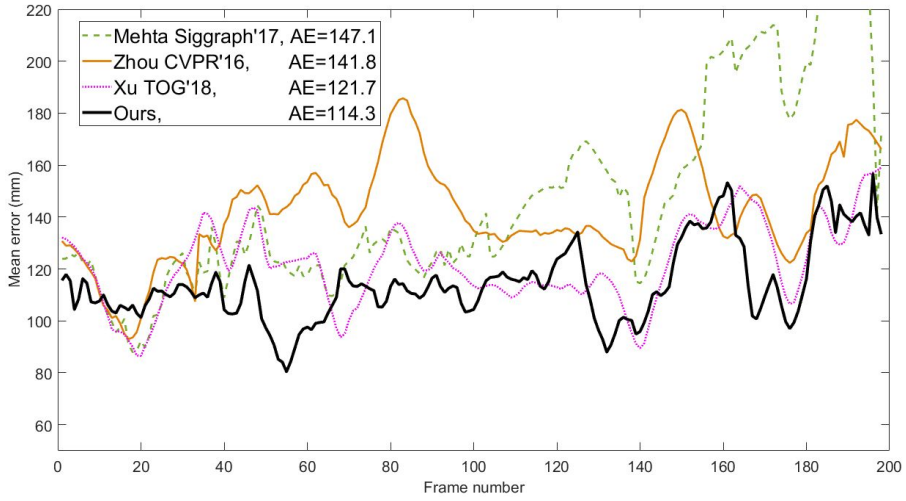


Figure 4.14: Comparison of the skeletal pose estimation accuracy in terms of average per-joint 3D error on the *Pablo* sequence. The proposed method outperforms the three other methods, most notably the skeleton-only methods of Mehta et al., 2017 and Zhou et al., 2016.

joint positions from the multi-view method of Robertini et al., 2016 are used. The average per-joint 3D error (in millimeters) is reported after aligning the per-frame poses with a similarity transform. As shown in Figure 4.14, the proposed method outperforms the three other methods, most notably the skeleton-only methods (Mehta et al., 2017; Zhou et al., 2016). This shows that the combined surface and skeleton reconstruction also benefits 3D pose estimation quality in itself.

4.8.7 *Ablation Study*

First, the importance of all algorithmic components is qualitatively evaluated in an ablation study on a real video sequence. To this end, the results of the proposed method are compared to: 1) the pose estimation without facial landmark alignment term and the silhouette term, which is referred to as $E_{2Dw/oface} + E_{3D}$, 2) the pose estimation without the silhouette term ($E_{2D} + E_{3D}$), 3) the complete pose estimation (E_{pose}) and 4) the full pipeline ($E_{pose} + E_{non-rigid}$). As shown in Figure 4.15, 1) the facial landmark alignment term significantly improves the head orientation estimation (red circles), 2) the misalignment of $E_{2D} + E_{3D}$ is corrected by the silhouette term in E_{pose} (yellow circles), 3) the non-rigid deformation on the surface, which cannot be modeled by skinning, is accurately captured by the proposed non-rigid registration method $E_{non-rigid}$ (blue circles). Second, the importance of the terms is also quantitatively evaluated on a sequence where high-quality reconstructions based on the multi-view performance capture results of De Aguiar et al., 2008 are used as ground truth. The mean vertex

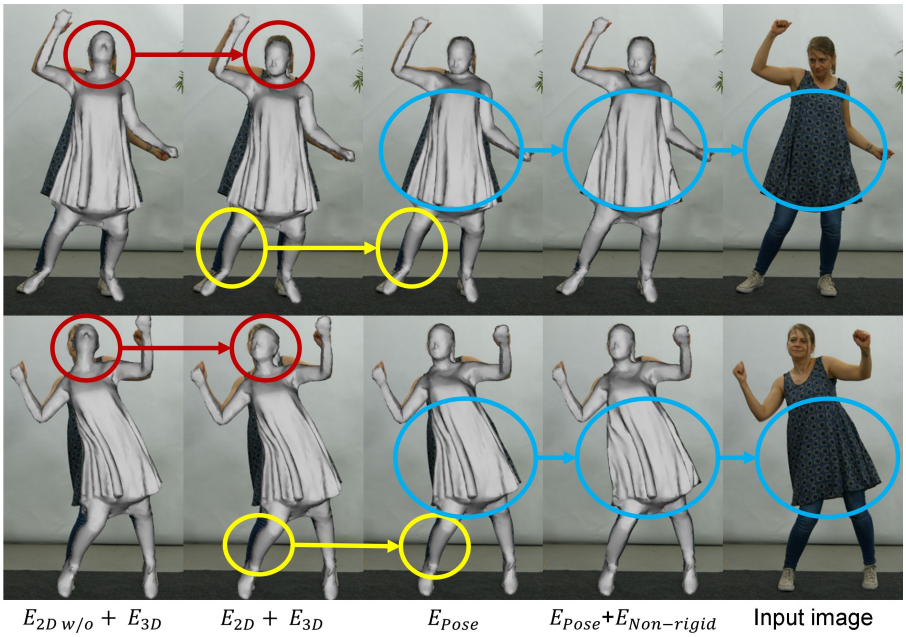


Figure 4.15: Ablation study. 1) the facial landmark alignment term significantly improves the head orientation estimation (red circles), 2) the misalignment of $E_{2D} + E_{3D}$ is corrected by the silhouette term in E_{pose} (yellow circles), 3) the non-rigid deformation on the surface, which cannot be modeled by skinning, is accurately captured by the non-rigid registration method $E_{non-rigid}$ (blue circles).

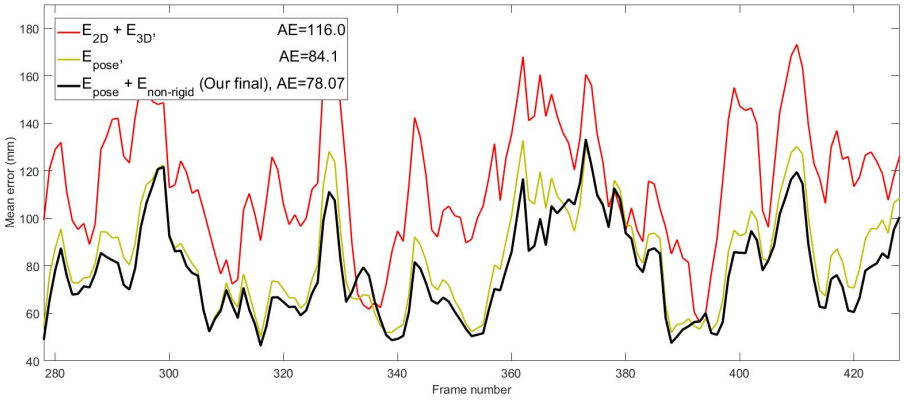


Figure 4.16: Ablation study. The mean vertex position error clearly demonstrates the consistent improvement by each of the algorithmic components of the approach. The full approach consistently obtains the lowest error.

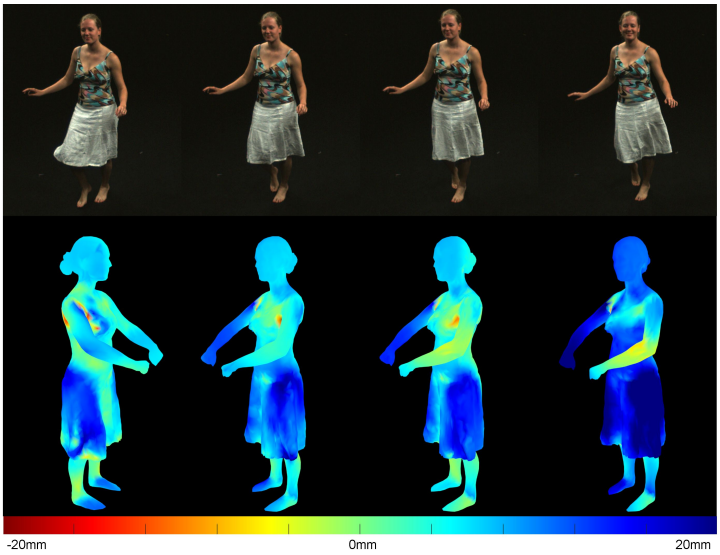


Figure 4.17: Improvement of the non-rigid stage ($E_{pose} + E_{non-rigid}$) over pose-only deformation (E_{pose}). Top row: Four monocular input images. On the bottom row, for each image, the per-vertex error of the pose only results minus the per-vertex error of the proposed method is shown. Consequently, a negative value means pose only is better, and it is colored in red. A positive value means the proposed method is better, and it is colored in blue. As expected, the presented approach achieves the most improvement on the non-rigid skirt part— which is around 20mm for the shown frames.

position error shown in Figure 4.16 clearly demonstrates the consistent improvement by each of the algorithmic components of the approach. The non-rigid alignment stage obtains, on average better results than the pose-only alignment. Since non-rigid deformations are most of the time concentrated in certain areas, e.g., a skirt, and at certain frames when articulated motion takes place, the per-frame and per-vertex improvement of the proposed non-rigid stage is also measured. To this end, the improvement of ($E_{pose} + E_{non-rigid}$) over (E_{pose}) is measured by computing the per-vertex error of the pose only results minus the per-vertex error of the proposed method. Consequently, positive means the presented approach is better than the pose-only deformation. As demonstrated in Figure 4.17, the non-rigid stage significantly improves the reconstruction of the skirt and the arm. The improvement is especially noticeable for frames where the deformation of the skirt significantly differs from the static template model since such motion cannot be handled by the pose-only step. On the same dataset, the influence of 1) the warping of the non-rigid displacement of the previous frame, 2) the proposed body part masks used in the dense silhouette alignment, and 3) the proposed vertex snapping are evaluated. Those algorithmic changes respectively lead to 2.4%, 1.7%, and 1.7% improvement in average 3D vertex error, which sums up to a total improvement of 5.8%. The importance of the material-based non-rigid deformation

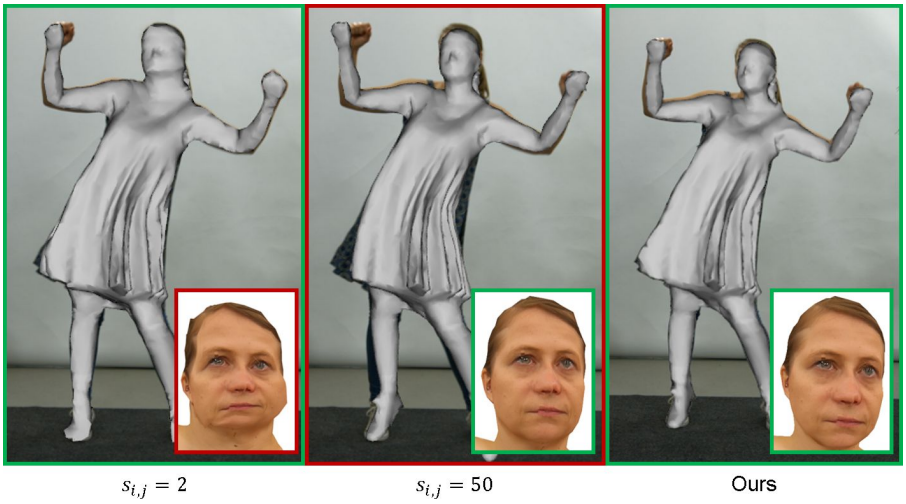


Figure 4.18: Importance of the material-based non-rigid deformation adaptation strategy. With a low global regularization weight, the deformation of the skirt is well reconstructed, but the head is distorted (left). A high deformation weight preserves the shape of the head but prevents tracking of the skirt motion (middle). The new semantic weight adaptation strategy enables the reconstruction of both regions with high accuracy and leads to the best results (right).

adaptation strategy is shown in Figure 4.18. Using constantly low non-rigidity weights ($s_{i,j} = 2.0$) in all regions, the deformation of the skirt is well reconstructed, but the head is severely distorted (left). In contrast, with high global non-rigidity weights ($s_{i,j} = 50.0$), the head shape is preserved, but the skirt cannot be tracked reliably (middle). The new semantic weight adaptation strategy enables the reconstruction of both regions with high accuracy and leads to the best results (right).

4.8.8 *Applications*

The proposed monocular real-time human performance capture method can facilitate many applications that depend on real-time capture: interactive VR and AR, human-computer interaction, pre-visualization for visual effects, 3D video, or telepresence. Two application demonstrators are exemplified here. In Figure 4.19, it is shown that the method allows live free-viewpoint video rendering and computer animation of the performance captured result from just single color input. This illustrates the potential of the presented method in several of the aforementioned live application domains. In Figure 4.20, a real-time virtual try-on application is demonstrated based on the proposed performance capture method. Here, the texture corresponding to the trousers on the template is exchanged, and the tracked result is visualized in real time. With such a system, the users can see themselves in clothing variants

Figure 4.19: Free-viewpoint video rendering results.

Figure 4.20: Live virtual try-on application based on the proposed approach.

in real time with live feedback, which could be potentially used in VR or even AR online shopping.

4.9 LIMITATIONS AND FUTURE WORK

Compelling real-time full-body human performance capture results have been demonstrated using only a single consumer-grade color camera. The proposed formulation combines constraints used individually in different image-based reconstruction methods before. However, the specific combination that is employed embedded in a hierarchical real-time approach is new and enables, for the first time, real-time monocular performance capture. Further, this formulation geared rigorously for real-time use differs from the related, but off-line MonoPerfCap (Xu et al., 2018) method in several ways: In Stage I, the facial landmarks as well as the displacement warping, which is also added during pose tracking, improve the pose accuracy of the real-time method. Further, the pose is tracked per frame instead of a batch-based formulation which reduces the computation time and allows faster motions. Further improvements in terms of efficiency are achieved by the GPU-based pose solver. In Stage II, the dense photometric term that adds con-

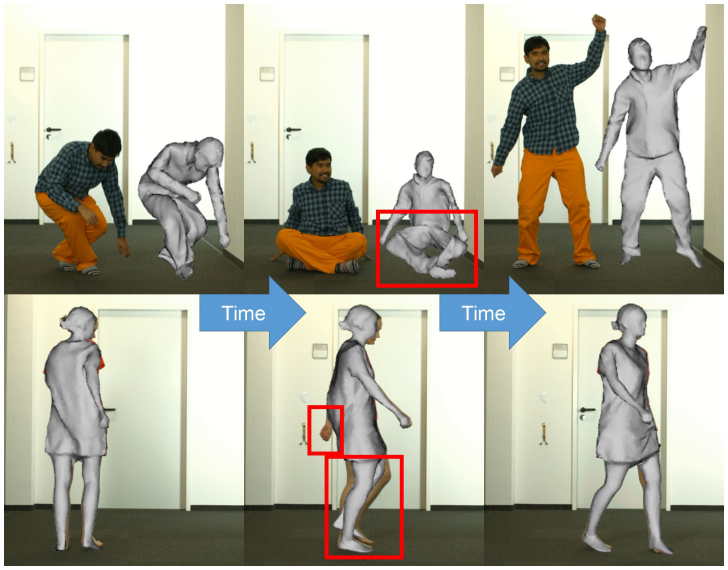


Figure 4.21: Failure cases. Top row: The underlying 3D joint regression deep network can fail for extreme poses not seen in training, which can produce glitches in the tracking results. The model fitting can often but not always correct such wrong estimates. However, the proposed performance capture approach robustly recovers from such situations. Bottom row: The estimates for occluded parts will be less accurate than with multi-view methods due to the lack of image evidence. While pose and silhouette plausibly constrain the back-side of the body, fully-occluded limbs may have incorrect poses.

straints for non-boundary vertices and the adaptive material-based regularization improve reconstruction quality. The non-rigid fitting stage is faster due to the more efficient combination of spatial regularizers that requires a much smaller number of variables than the as-rigid-as-possible regularizer. It is directly solved for the vertex displacements instead of estimating the embedded graph rotations/translations. This formulation is better suited for a parallel implementation on the GPU, and it also gives a more flexible representation. Due to the real-time constraint, an efficient distance transform-based representation is employed instead of the ICP-based approach that requires an expensive search of correspondences between the model boundary and the image silhouettes. The experiments show that the proposed method achieves a similar reconstruction quality compared to the off-line performance capture approach of Xu et al., 2018 while being orders of magnitude faster.

Nonetheless, the proposed approach is subject to some limitations; see also Figure 4.21. Due to the ambiguities that come along with monocular performance capture, the method relies on an accurate template acquisition since reconstruction errors and mislabeled part segmentations in the template itself cannot be recovered during tracking. Further, the method cannot handle topological changes that are too far from the template, e.g., removing some clothes and deformations

along the camera viewing axis can only be partially recovered by the photometric term. The latter point could be addressed by an additional term that involves shading and illumination estimation. As is common for learning methods, the underlying 3D joint regression deep network fails for extreme poses not seen in training. The model fitting can often, but not always, correct such wrong estimates, which produces glitches in the tracking results. However, the performance capture approach robustly recovers from such situations, see Figure 4.21 (top). Since the proposed method uses foreground/background segmentation, strong shadows and shading effects, objects with a similar color to the performer, and changing illumination situations can cause suboptimal segmentation; thus leading to noisy data association in the silhouette alignment term, which manifests itself as high-frequency jitter. The presented approach is robust to some degree of miss-classifications but can get confused by big segmentation outliers. This could be alleviated in the future by incorporating more sophisticated background segmentation strategies, e.g., based on deep neural networks. Strong changes in shading or shadows, specular materials, or non-diffuse lighting can also negatively impact the color alignment term. A joint optimization for scene illumination and material properties could alleviate this problem. Even though the components of the presented method are carefully orchestrated to achieve high accuracy and temporal stability in this challenging monocular setting, even under non-trivial occlusions, extensive (self-)occlusion is still fundamentally difficult. The estimates for occluded parts will be less accurate than with multi-view methods due to the lack of image evidence. While pose and silhouette plausibly constrain the back-side of the body, fully-occluded limbs may have incorrect poses. Additional learned motion priors could further resolve such ambiguous situations. Fortunately, the proposed approach recovers as soon as the difficult occlusions are gone, see Figure 4.21 (bottom).

4.10 CONCLUSION

This chapter of the thesis presented the first monocular real-time human performance capture approach that reconstructs dense, space-time coherent deforming geometry of entire humans in their loose everyday clothing. The novel energy formulation leverages automatically identified material regions on the template to differentiate between different non-rigid deformation behaviors of skin and various types of apparel. The underlying non-linear optimization problems are tackled in real time based on a pipelined implementation that runs two specially-tailored data-parallel Gauss-Newton solvers, one for pose estimation and one for non-rigid tracking, at the same time. The proposed approach can be seen as the first step towards general real-time capture of humans from just a single view, which is an invaluable tool

for believable, immersive virtual and augmented reality, telepresence, virtual try-on, and many more exciting applications the future will bring to our homes.

As stated in the limitations section, the approach presented in this chapter has a reduced 3D pose and surface accuracy in the case of occlusions. To this end, the next chapter proposes a fully learning-based method that regresses the pose and surface of a human from a single image while weak multi-view supervision during training improves the 3D pose and surface accuracy for both visible and occluded areas.

5

DEEPCAP: MONOCULAR HUMAN PERFORMANCE CAPTURE USING WEAK SUPERVISION

The previous chapter introduced a novel method for real-time human performance capture, which recovers the deforming surface of the entire human, including also the clothing, just using a single camera. Many previous performance capture approaches either require expensive multi-view setups, do not recover dense space-time coherent geometry with frame-to-frame correspondences, or have limited accuracy in 3D like the method proposed in the previous chapter. In this chapter, a novel deep learning approach (published as Habermann et al., 2020 and Habermann et al., 2021b) for monocular dense human performance capture is proposed, which aims at further improving the 3D performance by leveraging weak multi-view supervision during training. As the method is trained in a weakly supervised manner based on multi-view imagery, it completely removes the need for training data with 3D ground truth annotations. The network architecture is based on two separate networks that disentangle the task into a pose estimation and a non-rigid surface deformation step. Extensive qualitative and quantitative evaluations show that the approach outperforms the state of the art in terms of quality and robustness.

5.1 INTRODUCTION

Human performance capture, i.e., the space-time coherent 4D capture of full pose and non-rigid surface deformation of people in general clothing, revolutionized the film and gaming industry in recent years. Apart from visual effects, it has many use cases in generating personalized dynamic virtual avatars for telepresence, virtual try-on, mixed reality, and many other areas. In particular, for the latter applications, being able to performance capture humans from *monocular videos* would be a game-changer. The majority of established monocular methods only captures articulated motion (including hands or sparse facial expression at most). However, the monocular tracking of dense full-body deformations of skin and clothing, in addition to articulated pose, which plays an important role in producing realistic virtual characters, is still in its infancy.

In literature, multi-view marker-less methods (Bray et al., 2006; Brox et al., 2006, 2010; Cagniart et al., 2010; De Aguiar et al., 2008; Gall et al., 2009; Liu et al., 2011; Mustafa et al., 2015; Pons-Moll et al., 2017, 2015; Vlasic et al., 2008, 2009; Wu et al., 2013) have shown compelling results. However, these approaches rely on well-controlled

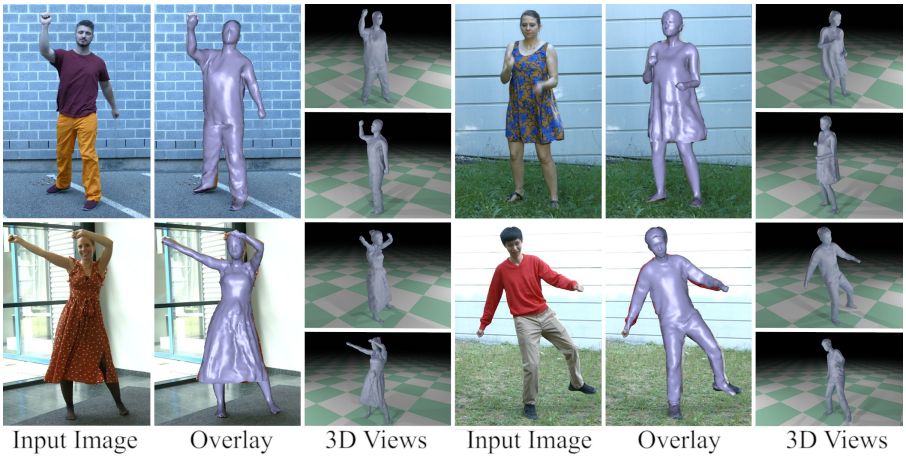


Figure 5.1: The first learning-based approach for dense monocular human performance capture using weak multi-view supervision is presented that predicts not only the pose but also the space-time coherent non-rigid deformations of the model surface.

multi-camera studios (typically with green screen), which prohibits them from being used for location shootings of films and telepresence in living spaces.

Recent monocular human modeling approaches have shown compelling reconstructions of humans, including clothing, hair, and facial details (Alldieck et al., 2019a, 2018b; Bhatnagar et al., 2019; Ma et al., 2020; Patel et al., 2020; Saito et al., 2019; Zheng et al., 2019). Some directly regress voxels (Gabeur et al., 2019; Zheng et al., 2019) or the continuous occupancy of the surface (Saito et al., 2019). Since predictions are pixel aligned, reconstructions have nice detail, but limbs are often missing, especially for difficult poses. Moreover, the recovered motion is not factorized into articulation and non-rigid deformation, which prevents the computer graphics style control over the reconstructions that is required in many of the aforementioned applications. Importantly, surface vertices are not tracked over time, so no space-time coherent model is captured. Another line of work predicts deformations or displacements to an articulated template, which prevents missing limbs and allows more control (Alldieck et al., 2019a,b; Bhatnagar et al., 2019; Pumarola et al., 2019). However, these works do not capture motion and surface deformations.

The state-of-the-art monocular human performance capture methods (Habermann et al., 2019; Xu et al., 2018) densely track the deformation of the surface. They leverage deep learning-based sparse keypoint detections and perform an expensive template fitting afterwards. In consequence, they can only non-rigidly fit the input view and suffer from instability. By contrast, the first learning-based method is presented that jointly infers the articulated and non-rigid 3D deformation

parameters in a single feed-forward pass at much higher performance, accuracy, and robustness (see also Figure 5.1). The core of the method is a CNN model which integrates a fully differentiable *mesh* template parameterized with *pose* and an *embedded deformation graph*. From a single image, the network predicts the skeletal pose and the rotation and translation parameters for each node in the deformation graph. In stark contrast to implicit representations (Chibane et al., 2020; Saito et al., 2019; Zheng et al., 2019), the proposed mesh-based method *tracks the surface vertices over time*, which is crucial for adding semantics and for texturing and rendering in graphics. Further, by virtue of the parameterization, the model always produces a human surface *without missing limbs*, even during occlusions and out-of-plane motions.

While previous methods (Alldieck et al., 2019a; Bhatnagar et al., 2019; Saito et al., 2019; Zheng et al., 2019) rely on 3D ground truth for training, the proposed method is weakly supervised from multi-view images. To this end, a fully differentiable architecture is proposed, which is trained in an analysis-by-synthesis fashion, without explicitly using any 3D ground truth annotation. Specifically, during training, the method only requires a personalized template mesh of the actor and a multi-view video sequence of the actor performing various motions. Then, the network learns to predict 3D pose and dense non-rigidly deformed surface shape by comparing its single image feed-forward predictions in a differentiable manner against the multi-view 2D observations. At test time, the proposed method only requires a single-view image as input and produces a deformed template matching the actor's non-rigid motion in the image. In summary, the main technical contributions are:

- A learning-based 3D human performance capture approach that jointly tracks the skeletal pose and the non-rigid surface deformations from monocular images.
- A new differentiable representation of deforming human surfaces, which enables training from multi-view video footage directly.

The new model achieves high-quality, dense human performance capture results on the new challenging dataset, demonstrating, qualitatively and quantitatively, the advantages of the proposed approach over previous work. It is experimentally shown that the method produces reconstructions of higher accuracy and 3D stability, in particular in depth, than related work, also under difficult poses.

5.2 OVERVIEW

Given a single RGB video of a moving human in general clothing and a respective template mesh (Section 5.3), the goal is to capture the dense

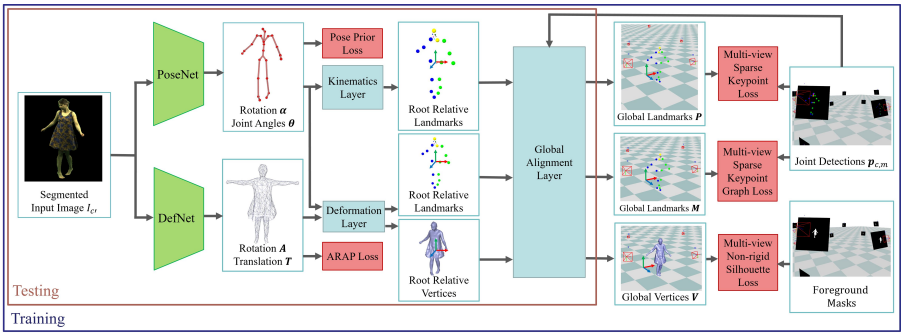


Figure 5.2: Overview of the proposed approach. The proposed method takes a single segmented image as input. First, the pose network, *PoseNet*, is trained to predict the joint angles and the camera relative rotation using sparse multi-view 2D joint detections as weak supervision. Second, the deformation network, *DefNet*, is trained to regress embedded graph rotation and translation parameters to account for non-rigid deformations. To train DefNet, multi-view 2D joint detections and silhouettes are used for supervision.

deforming surface of the full body. This is achieved by training a neural network consisting of two components: As illustrated in Figure 5.2, the pose network, *PoseNet*, estimates the skeletal pose of the actor in the form of joint angles from a monocular image (Section 5.5). Next, the deformation network, *DefNet*, regresses the non-rigid deformation of the dense surface, which cannot be modeled by the skeletal motion, in the embedded deformation graph representation (Section 5.6). To avoid generating dense 3D ground truth annotation, the network is trained in a weakly supervised manner. To this end, a fully differentiable human deformation and rendering model is proposed, which allows the proposed approach to compare the rendering of the human body model to the 2D image evidence and backpropagate the losses. For training, a video sequence in a calibrated multi-camera green screen studio is captured first (Section 5.4). Note that the multi-view video is only used during training. At test time, only a single RGB video and a dedicated domain adaptation step (Section 5.7) are required to perform dense non-rigid tracking.

5.3 CHARACTER MODEL

The proposed method relies on a person-specific 3D template model. First, the actor is scanned with a 3D scanner (*Treedys* 2020) to obtain the textured mesh. To create the textured mesh (see Figure 5.3), the person is captured in a static T-pose with an RGB-based scanner[1] which has 134 RGB cameras resulting in 134 images $\mathcal{I}_{\text{rec}} = \{I_{\text{rec}_1}, \cdots, I_{\text{rec}_{134}}\}$. The textured 3D geometry is obtained by leveraging a commercial 3D

1 https://www.treedys.com/

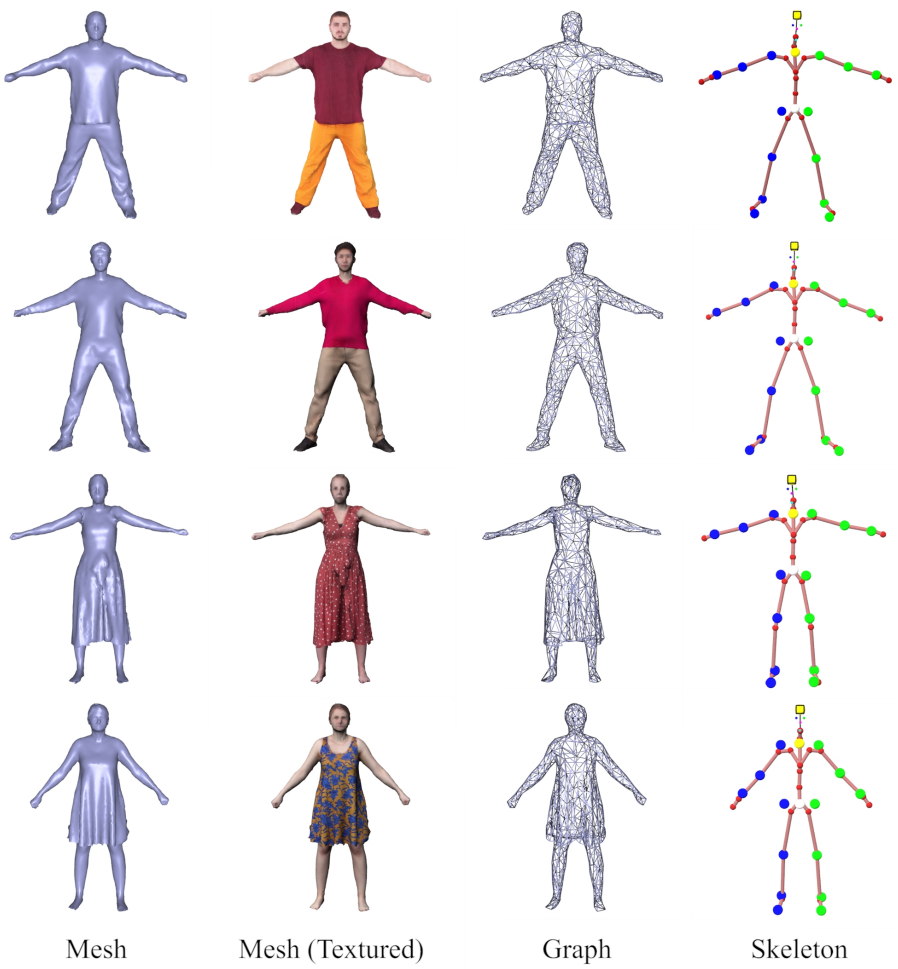


Figure 5.3: Character models. Here, the character model of *S*1 to *S*4 (top to bottom) of the new dataset is shown. It consists of the textured mesh, the underlying embedded deformation graph as well as the attached skeleton.

reconstruction software, called Agisoft Metashape[2], that takes as input the images $\mathcal{I}_{\text{rec}}$ and reconstructs a textured 3D mesh of the person (see Figure 5.3). Metashape's mesh simplification is applied to reduce the number of vertices N and Meshmixer's[3] remeshing is performed to obtain roughly uniform shaped triangular surfaces. Next, the skeleton (see Figure 5.3) is automatically fit to the 3D mesh by fitting the SMPL model (Loper et al., 2015). To this end, first, the pose is optimized by performing a sparse non-rigid ICP where the head, hands, and feet are used as feature points since they can be easily detected in a T-pose. Then, a dense non-rigid ICP is performed on vertex level to obtain the final pose and shape parameters. For clothing types that roughly follow the human body shape, e.g., pants and shirt, the per-vertex skinning weights of the naked SMPL model are propagated to the template vertices. For other types of clothing, like skirts and dresses, Blenders's[4] automated skinning weight computation is leveraged. The skeleton consists of 23 joints and 21 attached landmarks (17 body and 4 face landmarks) and is parameterized with 27 joint angles $\boldsymbol{\theta} \in \mathbb{R}^{27}$, the camera relative rotation $\boldsymbol{\alpha} \in \mathbb{R}^3$ and translation $\mathbf{t} \in \mathbb{R}^3$. The landmark placement follows the convention of OpenPose (Cao et al., 2018, 2017; Simon et al., 2017; Wei et al., 2016). To model the non-rigid surface deformation, an embedded deformation graph $\mathcal{G}$ with K nodes is automatically build by further decimating the template mesh to around 500 vertices (see Figure 5.3). The connections of a node k to neighboring nodes are given by the vertex connections of the decimated mesh and are denoted as the set $\mathcal{N}_{\text{n}}(k)$. For each vertex of the decimated mesh, it is searched for the closest vertex on the template mesh in terms of Euclidean distance. These points then define the position of the graph nodes $\mathbf{G} \in \mathbb{R}^{K\times 3}$ where $\mathbf{G}_k$ is the position of node k. To compute the vertex-to-node weights $w_{i,k}$, the geodesic distance is measured between the graph node k and the template vertex i, and $\mathcal{N}_{\text{vn}}(i)$ denotes the set of nodes that influence vertex i. The nodes are parameterized with Euler angles $\mathbf{A} \in \mathbb{R}^{K\times 3}$ and translations $\mathbf{T} \in \mathbb{R}^{K\times 3}$. Similar to (Habermann et al., 2019), the mesh is segmented into different non-rigidity classes resulting in per-vertex rigidity weights s_i. This enables the modeling of varying deformation behaviors for different surface materials, e.g., skin deforms less than clothing (see Equation 5.13).

5.4 TRAINING DATA

To acquire the training data, a multi-view video of the actor doing various actions is recorded in a calibrated multi-camera studio with a green screen. The number of frames per subject varies between 26,000 and 38,000 depending on how fast the person performed all the motions. *C*

2 http://www.agisoft.com
3 http://www.meshmixer.com/
4 https://www.blender.org/

calibrated and synchronized cameras with a resolution of 1024×1024 are leveraged for capturing where for all subjects between 11 and 14 cameras are used. To provide weak supervision for the training, first, 2D pose detection is performed on the sequences using OpenPose (Cao et al., 2018, 2017; Simon et al., 2017; Wei et al., 2016), and temporal filtering is applied. Then, the foreground mask is generated using color keying and the corresponding distance transform image $D_{f,c}$ (Borgefors, 1986) is computed, where $f \in [0, F]$ and $c \in [0, C]$ denote the frame index and camera index, respectively. The original image resolution is too large to transfer all the distance transform images to the GPU during training. Fortunately, most of the image information is anyways redundant since one is only interested in the image region where the person is. Therefore, the distance transform images are cropped using the bounding box that contains the segmentation mask with a conservative margin. Finally, they are resized to a resolution of 350×350 without losing important information. During training, one camera view c' and frame f' is randomly sampled for which the recorded image is cropped with a bounding box, based on the 2D joint detections. The final training input image $I_{f',c'} \in \mathbb{R}^{256 \times 256 \times 3}$ is obtained by removing the background and augmenting the foreground with random brightness, hue, contrast, and saturation changes. For simplicity, the operation on frame f' is described, and the subscript f' is omitted in the following equations.

5.5 POSE NETWORK

In the proposed *PoseNet*, ResNet50 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009) is used as a backbone and the last fully connected layer is modified to output a vector containing the joint angles $\boldsymbol{\theta}$ and the camera relative root rotation $\boldsymbol{\alpha}$, given the input image $I_{c'}$. Since generating the ground truth for $\boldsymbol{\theta}$ and $\boldsymbol{\alpha}$ is a non-trivial task, a weakly supervised training is proposed based on fitting the skeleton to multi-view 2D joint detections.

5.5.1 *Kinematics Layer*

To this end, a kinematics layer is introduced as the differentiable function that takes the joint angles $\boldsymbol{\theta}$ and the camera relative rotation $\boldsymbol{\alpha}$ and computes the positions $\mathbf{P}_{c'} \in \mathbb{R}^{M \times 3}$ of the M 3D landmarks attached to the skeleton (17 body joints and 4 face landmarks). Note that $\mathbf{P}_{c'}$ lives in a camera-root-relative coordinate system. In order to project the landmarks to other camera views, $\mathbf{P}_{c'}$ has to be transformed to the world coordinate system:

$$\mathbf{P}_m = \mathbf{R}_{c'}^T \mathbf{P}_{c',m} + \mathbf{t}, \tag{5.1}$$

where $\mathbf{R}_{c'}$ is the rotation matrix of the input camera c' and $\mathbf{t}$ is the global translation of the skeleton.

5.5.2 *Global Alignment Layer*

To obtain the global translation $\mathbf{t}$, a global alignment layer is proposed that is attached to the kinematics layer. It localizes the skeleton model in the world space, such that the globally rotated landmarks $\mathbf{R}_{c'}^T\mathbf{P}_{c',m}$ project onto the corresponding detections in all camera views. This is done by minimizing the distance between the rotated landmarks $\mathbf{R}_{c'}^T\mathbf{P}_{c',m}$ and the corresponding rays cast from the camera origin $\mathbf{o}_c$ to the 2D joint detections:

$$\sum_c \sum_m \sigma_{c,m} \| (\mathbf{R}_{c'}^T\mathbf{P}_{c',m} + \mathbf{t} - \mathbf{o}_c) \times \mathbf{d}_{c,m} \|^2, \tag{5.2}$$

where $\mathbf{d}_{c,m}$ is the direction of a ray from camera c to the 2D joint detection $\mathbf{p}_{c,m}$ corresponding to landmark m:

$$\mathbf{d}_{c,m} = \frac{(\mathbf{E}_c^{-1}\tilde{\mathbf{p}}_{c,m})_{xyz} - \mathbf{o}_c}{\| (\mathbf{E}_c^{-1}\tilde{\mathbf{p}}_{c,m})_{xyz} - \mathbf{o}_c \|}. \tag{5.3}$$

Here, $\mathbf{E}_c \in \mathbb{R}^{4\times4}$ is the projection matrix of camera c and $\tilde{\mathbf{p}}_{c,m} = (\mathbf{p}_{c,m}, 1, 1)^T$. Each point-to-line distance is weighted by the joint detection confidence $\sigma_{c,m}$, which is set to zero if below 0.4. The minimization problem of Equation 5.2 can be solved in closed form:

$$\mathbf{t} = \mathbf{W}^{-1} \sum_{c,m} \mathbf{D}_{c,m}(\mathbf{R}_{c'}^T\mathbf{P}_{c',m} - \mathbf{o}_c) + \mathbf{o}_c - \mathbf{R}_{c'}^T\mathbf{P}_{c',m}, \tag{5.4}$$

where

$$\mathbf{W} = \sum_c \sum_m \mathbf{I} - \mathbf{D}_{c,m}. \tag{5.5}$$

Here, $\mathbf{I}$ is the 3×3 identity matrix and $\mathbf{D}_{c,m} = \mathbf{d}_{c,m}\mathbf{d}_{c,m}^T$. Note that the operation in Equation 5.4 is differentiable with respect to the landmark positions $\mathbf{P}_{c'}$.

5.5.3 *Sparse Keypoint Loss*

The 2D sparse keypoint loss for the *PoseNet* can be expressed as

$$\mathcal{L}_{\text{kp}}(\mathbf{P}) = \sum_c \sum_m \lambda_m \sigma_{c,m} \| \pi_c(\mathbf{P}_m) - \mathbf{p}_{c,m} \|^2, \tag{5.6}$$

which ensures that each landmark projects onto the corresponding 2D joint detections $\mathbf{p}_{c,m}$ in all camera views. Here, π_c is the projection function of camera c and $\sigma_{c,m}$ is the same as in Equation 5.2. λ_m is a hierarchical re-weighting factor that varies during training for better convergence. More precisely, for the first one-third of the training

iterations per training stage for *PoseNet*, the keypoint loss is multiplied with a factor of $\lambda_m = 3$ for all torso markers and with a factor of $\lambda_m = 2$ for elbow and knee markers. For all other markers, lambda is set to $\lambda_m = 1$. For the remaining iterations, it is set to $\lambda_m = 3$ for all markers. This re-weighting lets the model first focus on the global rotation (by weighting torso markers higher than others). It is found that this gives better convergence during training and joint angles overshoot less often, especially at the beginning of training.

5.5.4 *Pose Prior Loss*

To avoid unnatural poses, a pose prior loss is imposed on the joint angles

$$\mathcal{L}_{\text{limit}}(\boldsymbol{\theta}) = \sum_{i=1}^{27} \Psi(\boldsymbol{\theta}_i) \tag{5.7}$$

$$\Psi(x) = \begin{cases} (x - \boldsymbol{\theta}_{\text{max},i})^2, \text{ if } x > \boldsymbol{\theta}_{\text{max},i} \\ (\boldsymbol{\theta}_{\text{min},i} - x)^2, \text{ if } x < \boldsymbol{\theta}_{\text{min},i} \\ 0 \qquad\qquad\quad , \text{ otherwise} \end{cases}, \tag{5.8}$$

that encourages that each joint angle $\boldsymbol{\theta}_i$ stays in a range $[\boldsymbol{\theta}_{\text{min},i}, \boldsymbol{\theta}_{\text{max},i}]$ depending on the anatomic constraints.

5.6 DEFORMATION NETWORK

With the skeletal pose from *PoseNet* alone, the non-rigid deformation of the skin and clothes cannot be fully explained. Therefore, the non-rigid deformation and the articulated skeletal motion are disentangled. *DefNet* takes the input image $I_{c'}$ and regresses the non-rigid deformation parameterized with rotation angles $\mathbf{A}$ and translation vectors $\mathbf{T}$ of the nodes of the embedded deformation graph. *DefNet* uses the same backbone architecture as *PoseNet*, while the last fully connected layer outputs a $6K$-dimensional vector reshaped to match the dimensions of $\mathbf{A}$ and $\mathbf{T}$. The weights of *PoseNet* are fixed while training *DefNet*. Again, no direct supervision is used for $\mathbf{A}$ and $\mathbf{T}$. Instead, a deformation layer with differentiable rendering is proposed, and a multi-view silhouette-based weak supervision is leveraged.

5.6.1 *Deformation Layer*

The deformation layer takes $\mathbf{A}$ and $\mathbf{T}$ from *DefNet* as input to non-rigidly deform the surface

$$\mathbf{Y}_i = \sum_{k \in \mathcal{N}_{\text{vn}}(i)} w_{i,k}(R(\mathbf{A}_k)(\hat{\mathbf{V}}_i - \mathbf{G}_k) + \mathbf{G}_k + \mathbf{T}_k). \tag{5.9}$$

Here, $\mathbf{Y}, \hat{\mathbf{V}} \in \mathbb{R}^{N\times 3}$ are the vertex positions of the deformed and undeformed template mesh, respectively. $w_{i,k}$ are vertex-to-node weights, but in contrast to (Sumner et al., 2007), they are computed based on geodesic distances. $\mathbf{G} \in \mathbb{R}^{K\times 3}$ are the node positions of the undeformed graph, $\mathcal{N}_{\text{vn}}(i)$ is the set of nodes that influence vertex i, and $R(\cdot)$ is a function that converts the Euler angles to rotation matrices. Further, the skeletal pose is applied on the deformed mesh vertices to obtain the vertex positions in the input camera space

$$\mathbf{V}_{c',i} = \sum_{k\in\mathcal{N}_{\text{vn}}(i)} w_{i,k}(R_{\text{sk},k}(\boldsymbol{\theta},\boldsymbol{\alpha})\mathbf{Y}_i + t_{\text{sk},k}(\boldsymbol{\theta},\boldsymbol{\alpha})), \tag{5.10}$$

where the node rotation $R_{\text{sk},k}$ and translation $t_{\text{sk},k}$ are derived from the pose parameters using dual quaternion skinning (Kavan et al., 2007). Equation 5.9 and Equation 5.10 are differentiable with respect to pose and graph parameters. Thus, the layer can be integrated into the learning framework, and gradients can be propagated to *DefNet*. So far, $\mathbf{V}_{c',i}$ is still rotated relative to the camera c' and located around the origin. To bring them to global space, the inverse camera rotation and the global translation are applied, defined in Equation 5.4, $\mathbf{V}_i = \mathbf{R}_{c'}^T\mathbf{V}_{c',i} + \mathbf{t}$.

5.6.2 *Non-rigid Silhouette Loss*

This loss encourages that the non-rigidly deformed mesh matches the multi-view silhouettes in all camera views. It can be formulated using the distance transform representation (Borgefors, 1986)

$$\mathcal{L}_{\text{sil}}(\mathbf{V}) = \sum_c \sum_{i\in\mathcal{B}_c} \rho_{c,i} \| D_c\left(\pi_c\left(\mathbf{V}_i\right)\right) \|^2. \tag{5.11}$$

Here, $\mathcal{B}_c$ is the set of vertices that lie on the boundary when the deformed 3D mesh is projected onto the distance transform image D_c of camera c. Those vertices are computed by rendering a depth map using a custom CUDA-based rasterizer that can be easily integrated into deep learning architectures as a separate layer. The vertices that project onto a depth discontinuity (background vs. foreground) in the depth map are treated as boundary vertices. $\rho_{c,i}$ is a directional weighting (Habermann et al., 2019) that guides the gradient in D_c. The silhouette loss ensures that the boundary vertices project onto the zero-set of the distance transform, i.e., the foreground silhouette.

5.6.3 *Sparse Keypoint Graph Loss*

Only using the silhouette loss can lead to wrong mesh-to-image assignments, especially for highly articulated motions. To this end, a

sparse keypoint loss is used to constrain the mesh deformation, which is similar to the keypoint loss for *PoseNet* in Equation 5.6

$$\mathcal{L}_{\text{kpg}}(\mathbf{M}) = \sum_{c}\sum_{m}\sigma_{c,m}\|\pi_c\left(\mathbf{M}_m\right) - \mathbf{p}_{c,m}\|^2. \tag{5.12}$$

Differently from Equation 5.6, the deformed and posed landmarks $\mathbf{M}$ are derived from the embedded deformation graph. To this end, the canonical landmark positions can be deformed and posed by attaching them to its closest graph node g in canonical pose with weight $w_{m,g} = 1.0$. Landmarks can then be deformed according to Equation 5.9, 5.10, resulting in $\mathbf{M}_{c'}$ which is brought to global space via $\mathbf{M}_m = \mathbf{R}_{c'}^T\mathbf{M}_{c',m} + \mathbf{t}$.

5.6.4 *As-rigid-as-possible Loss*

To enforce local smoothness of the surface, an as-rigid-as-possible loss (Sorkine and Alexa, 2007) is imposed

$$\mathcal{L}_{\text{arap}}(\mathbf{A},\mathbf{T}) = \sum_{k}\sum_{l\in\mathcal{N}_{\text{n}}(k)} u_{k,l}\|d_{k,l}(\mathbf{A},\mathbf{T})\|_1, \tag{5.13}$$

where

$$d_{k,l}(\mathbf{A},\mathbf{T}) = R(\mathbf{A}_k)(\mathbf{G}_l - \mathbf{G}_k) + \mathbf{T}_k + \mathbf{G}_k - (\mathbf{G}_l + \mathbf{T}_l).$$

$\mathcal{N}_{\text{n}}(k)$ is the set of indices of the neighbors of node k. In contrast to (Sorkine and Alexa, 2007), weighting factors $u_{k,l}$ are proposed that influence the rigidity of respective parts of the graph. $u_{k,l}$ can be derived by averaging all per-vertex rigidity weights s_i (Habermann et al., 2019) for all vertices (see Section 5.3), which are connected to node k or l. Thus, the mesh can deform either less or more depending on the surface material. For example, graph nodes that are mostly connected to vertices on a skirt can deform more freely than nodes that are mainly connected to vertices on the skin.

5.7 IN-THE-WILD DOMAIN ADAPTATION

Since the training set is captured in a green screen studio and the test set is captured in the wild, there is a significant domain gap between them due to different lighting conditions and camera response functions. To improve the performance of the proposed method on in-the-wild images, the networks are fine-tuned on the monocular test images for a small number of iterations using the same 2D keypoint and silhouette losses as before, *but only on a single view*. This drastically improves the performance at test time, as shown in Figure 5.12.

Figure 5.4: Qualitative results. Each row shows results for a different person with varying types of apparel. The input frames and the reconstruction overlayed to the corresponding frame are visualized. Note that the results precisely overlay the input. Further, the reconstructions are shown from a virtual 3D viewpoint. Note that they also look plausible in 3D.

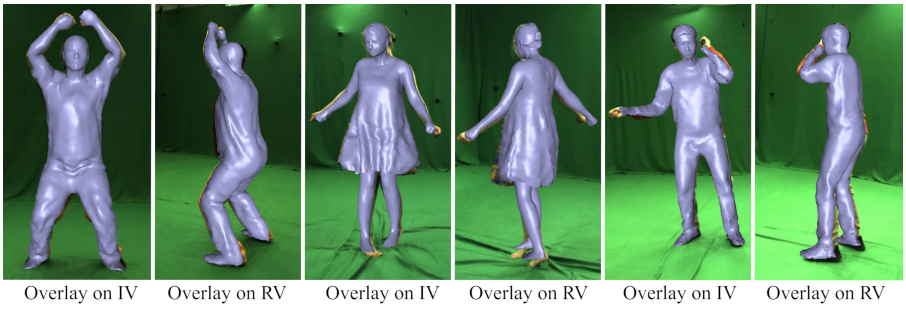


Figure 5.5: Results on the evaluation sequences where input views (IV) and reference views (RV) are available. Note that the reconstruction also precisely overlays on RV even though they are not used for tracking.

5.8 EVALUATION

All experiments were performed on a machine with an NVIDIA Tesla V100 GPU. A forward pass of the proposed method takes around 50ms, which breaks down to 25ms for *PoseNet* and 25ms for *DefNet*. During testing, the off-the-shelf video segmentation method of (Caelles et al., 2017) is used to remove the background in the input image. The proposed method requires OpenPose's 2D joint detections (Cao et al., 2018, 2017; Simon et al., 2017; Wei et al., 2016) as input during testing to crop the frames and to obtain the 3D global translation with the global alignment layer. Finally, the output mesh vertices are temporally smoothed with a Gaussian kernel of size 5 frames.

5.8.1 *Dataset*

The proposed approach is evaluated on 4 subjects (*S1* to *S4*) with varying types of apparel. For qualitative evaluation, 13 in-the-wild sequences are recorded in different indoor and outdoor environments shown in Figure 5.4. For quantitative evaluation, 4 sequences in a calibrated multi-camera green screen studio are captured (see Figure 5.5), for which the ground truth 3D joint locations are calculated using the multi-view motion capture software, The Captury (*The Captury* 2020), and a color keying algorithm is used for ground truth foreground segmentation. All sequences contain a large variety of motions, ranging from simple ones like walking up to more difficult ones like fast dancing or baseball pitching. The dataset is released for future research.

5.8.2 *Qualitative Comparisons*

Figure 5.4 shows qualitative results on in-the-wild test sequences with various clothing styles, poses, and environments. The reconstructions not only precisely overlay with the input images but also look plausible from arbitrary 3D viewpoints. In Figure 5.6 and 5.7, the approach is qualitatively compared to the related human capture and reconstruction methods (Habermann et al., 2019; Kanazawa et al., 2018; Saito et al., 2019; Zheng et al., 2019) on the green screen and the in-the-wild sequences, respectively. In terms of the shape representation, the proposed method is most closely related to LiveCap (Habermann et al., 2019) that also uses a person-specific template. Since they non-rigidly fit the template only to the monocular input view, their results do not faithfully depict the deformation in other viewpoints. Further, their pose estimation severely suffers from the monocular ambiguities, whereas the pose results of the proposed method are more robust and accurate. Comparing to the other three methods (Kanazawa et al.,

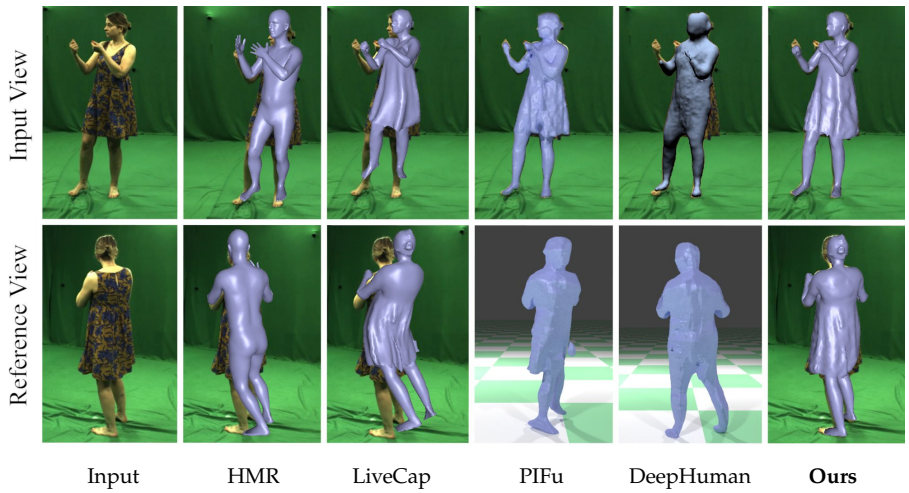


Figure 5.6: Qualitative comparison to other methods (Habermann et al., 2019; Kanazawa et al., 2018; Saito et al., 2019; Zheng et al., 2019) on the green screen evaluation sequences. Note that the results of the proposed approach overlay more accurately to the input view and also look more plausible from a reference view that was not used for tracking. Ground truth global translation is used to match the reference view for the results of (Habermann et al., 2019; Kanazawa et al., 2018). Since PIFu (Saito et al., 2019) and DeepHuman (Zheng et al., 2019) output meshes with varying topology in a canonical volume without an attached root, it is not possible to apply the ground truth translation, and therefore the reference view is shown without overlay.

2018; Saito et al., 2019; Zheng et al., 2019) that are trained for general subjects, the presented approach has the following advantages: First, the method recovers the non-rigid deformations of humans in general clothes, whereas the parametric model-based approaches (Kanazawa et al., 2018, 2019) only recover naked body shape. Second, the method directly provides surface correspondences over time which is important for AR/VR applications. In contrast, the results of implicit representation-based methods, PIFu (Saito et al., 2019) and DeepHuman (Zheng et al., 2019), lack temporal surface correspondences and do not preserve the skeletal structure of the human body, i.e., they often exhibit missing arms and disconnected geometry. Furthermore, DeepHuman (Zheng et al., 2019) only recovers a coarse shape in combination with a normal image of the input view, while the proposed method can recover medium-level detailed geometry that looks plausible from all views. Last but not least, all these existing methods have problems when overlaying their reconstructions on the reference view, even though some of the methods show a very good overlay on the input view. In contrast, the presented approach reconstructs accurate 3D geometry, and therefore, the results can precisely overlay on the reference views (also see Figure 5.5, 5.8, 5.9, and 5.10).

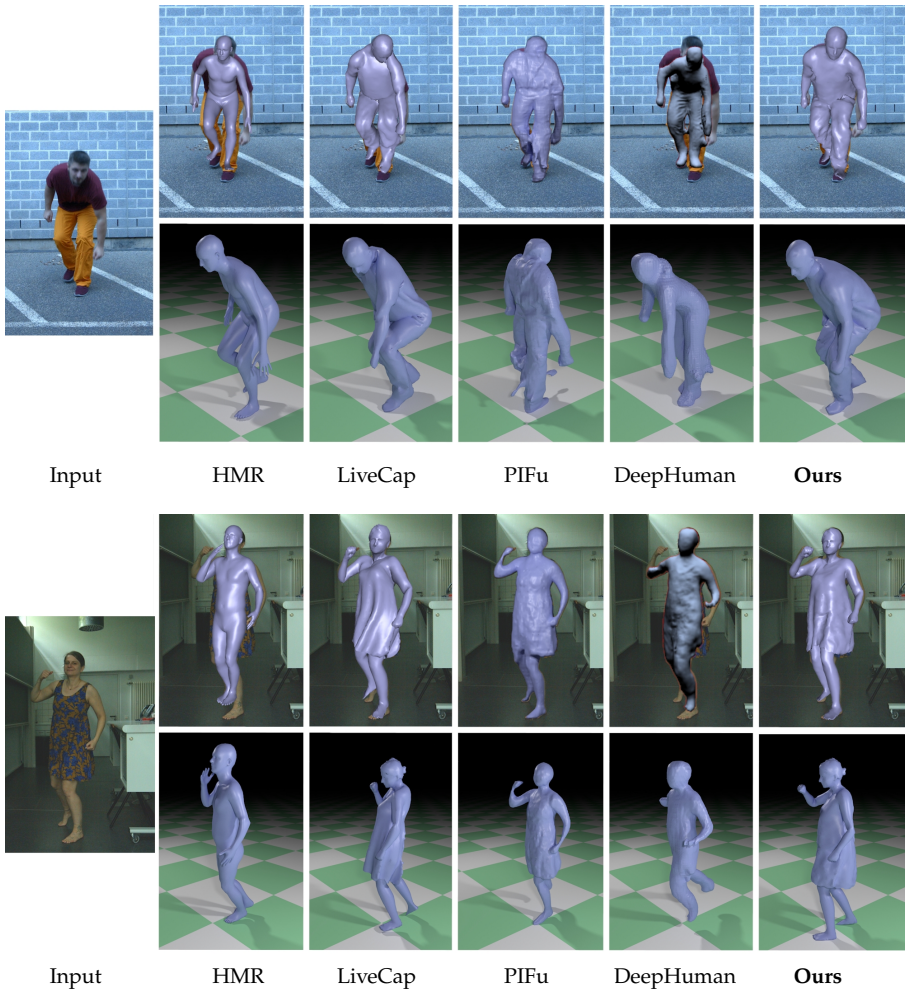


Figure 5.7: Comparisons to related work (Habermann et al., 2019; Kanazawa et al., 2018; Saito et al., 2019; Zheng et al., 2019) on the in-the-wild sequences showing *S1* and *S4*. The proposed approach can recover the deformations of clothing in contrast to (Kanazawa et al., 2018) and gives more stable and accurate results in 3D compared to (Habermann et al., 2019). Moreover, note that in contrast to previous work (Saito et al., 2019; Zheng et al., 2019), the presented method regresses space-time coherent geometry, which follows the structure of the human body.

5.8.3 *Skeletal Pose Accuracy*

The pose results (output of *PoseNet*) are quantitatively compared to existing pose estimation methods on *S1* to *S4*. To account for different types of apparel, *S1* and *S2* are chosen who wear trousers and a T-shirt or a pullover as well as *S3* and *S4* who wear a long and short dress, respectively. The bone lengths are rescaled for all methods to the ground truth, and the following metrics are evaluated on the 14 commonly used joints (Mehta et al., 2017) for every 10th frame: 1) The root joint position error or global localization error (*GLE*) is evaluated to measure how well the skeleton is placed in global 3D space. Note that *GLE* can only be evaluated for LiveCap (Habermann et al., 2019) and the presented approach since other methods only produce up-to-scale depth. 2) To evaluate the accuracy of the pose estimation, the 3D percentage of correct keypoints (3DPCK) with a threshold of 150*mm* of the root aligned poses and the area under the 3DPCK curve (AUC) are reported. 3) To factor out the errors in the global rotation; also the mean per joint position error (MPJPE) after Procrustes alignment is reported. The proposed approach is compared against the state-of-the-art pose estimation approaches, including VNect (Mehta et al., 2017), HMR (Kanazawa et al., 2018), HMMR (Kanazawa et al., 2019), and LiveCap (Habermann et al., 2019). It is also compared to a multi-view baseline approach (*MVBL*), where the differentiable skeleton model is used in an optimization framework to solve for the pose per frame using the proposed multi-view losses. One can see from Table 5.3 that the presented approach outperforms the related monocular methods in all metrics by a large margin and is even close to *MVBL* although the proposed method only takes a single image as input. The proposed method is further compared to VNect (Mehta et al., 2017) fine-tuned on the training images for *S1*. To this end, the 3D joint position is computed using The Captury (*The Captury* 2020) to provide ground truth supervision for VNect. On the evaluation sequence for *S1*, the fine-tuned VNect achieved 95.66% 3DPCK, 52.13% AUC and 47.16*mm* MPJPE. This shows that the presented weakly supervised approach yields comparable or better results than supervised methods in the person-specific setting. However, the proposed approach does not require 3D ground truth annotation that is difficult to obtain, even for only sparse keypoints, let alone the dense surfaces. Further note that even for *S3* accurate results can be achieved even though she wears a long dress such that legs are mostly occluded. On *S2*, it is found that the results of the presented approach are more accurate than *MVBL* since the classical frame-to-frame optimization can get stuck in local minima, leading to wrong poses.

MPJPE/GLE (in mm) and 3DPCK/AUC (in %) on S1				
Method	**GLE↓**	**3DPCK↑**	**AUC↑**	**MPJPE↓**
VNect (Mehta et al., 2017)	-	66.06	28.02	77.19
HMR (Kanazawa et al., 2018)	-	82.39	43.61	72.61
HMMR (Kanazawa et al., 2019)	-	87.48	45.33	72.40
LiveCap (Habermann et al., 2019)	317.01	71.13	37.90	92.84
Ours	**91.08**	**98.43**	**58.71**	**49.11**
MVBL	76.03	99.17	57.79	45.44

MPJPE/GLE (in mm) and 3DPCK/AUC (in %) on S2				
Method	**GLE↓**	**3DPCK↑**	**AUC↑**	**MPJPE↓**
VNect (Mehta et al., 2017)	-	80.50	39.98	66.96
HMR (Kanazawa et al., 2018)	-	80.02	39.24	71.87
HMMR (Kanazawa et al., 2019)	-	82.08	41.00	74.69
LiveCap (Habermann et al., 2019)	142.39	79.17	42.59	69.18
Ours	**75.79**	**94.72**	**54.61**	**52.71**
MVBL	64.12	89.91	45.58	57.52

MPJPE/GLE (in mm) and 3DPCK/AUC (in %) on S3				
Method	**GLE↓**	**3DPCK↑**	**AUC↑**	**MPJPE↓**
VNect (Mehta et al., 2017)	-	78.03	41.95	88.14
HMR (Kanazawa et al., 2018)	-	83.37	42.37	79.02
HMMR (Kanazawa et al., 2019)	-	79.93	36.27	91.62
LiveCap (Habermann et al., 2019)	281.27	66.30	31.44	98.76
Ours	**89.54**	**95.09**	**54.00**	**58.77**
MVBL	67.82	96.37	54.99	56.08

MPJPE/GLE (in mm) and 3DPCK/AUC (in %) on S4				
Method	**GLE↓**	**3DPCK↑**	**AUC↑**	**MPJPE↓**
VNect (Mehta et al., 2017)	-	82.06	42.73	72.62
HMR (Kanazawa et al., 2018)	-	86.88	43.91	73.63
HMMR (Kanazawa et al., 2019)	-	82.80	41.18	77.41
LiveCap (Habermann et al., 2019)	248.67	75.11	37.35	83.48
Ours	**96.56**	**96.74**	**59.25**	**45.40**
MVBL	75.82	96.20	57.27	45.12

Table 5.1: Skeletal pose accuracy. Note that the proposed approach is consistently better than other monocular approaches. Moreover, it is even close to the multi-view baseline.

5.8.4 *Surface Reconstruction Accuracy*

To evaluate the accuracy of the regressed non-rigid deformations, the intersection over union (IoU) is computed between the ground truth foreground masks and the 2D projection of the estimated shape on *S1* and *S4* for every 100th frame. The IoU is evaluated on *all views*, on *all views except the input view*, and on the *input view* which is referred to as *AMVIoU*, *RVIoU* and *SVIoU*, respectively. To factor out the errors in global localization, the ground truth translation is applied to the reconstructed geometries. For DeepHuman (Zheng et al., 2019) and PIFu (Saito et al., 2019), the *AMVIoU* and *RVIoU* cannot be reported since one cannot overlay their results on reference views as discussed before. Further, PIFu (Saito et al., 2019) by design achieves perfect overlay on the input view since they regress the depth for each foreground pixel. However, their reconstruction does not reflect the true 3D geometry (see Figure 5.6). Therefore, it is meaningless to report their *SVIoU*. Similarly, DeepHuman (Zheng et al., 2019) achieves high *SVIoU* due to their volumetric representation. But their results are often wrong when looking from side views. In contrast, DeepCap consistently outperforms all other approaches in terms of *AMVIoU* and *RVIoU*, which shows the high accuracy of the proposed method in recovering the 3D geometry. Further, the results are again close to the multi-view baseline.

5.8.5 *Ablation Study*

To evaluate the importance of the number of cameras, the number of training images, and the *DefNet*, an ablation study is performed on *S4* in Table 5.3. 1) In the first group of Table 5.3, the proposed networks are trained with supervision using 1 to 14 views. One can see that adding more views consistently improves the quality of the estimated poses and deformations. The most significant improvement is from one to two cameras. This is not surprising since the single-camera setting is inherently ambiguous. In Figure 5.8, the importance of the number of cameras is also shown qualitatively. 2) In the second group of Table 5.3 and in Figure 5.9, the training data is reduced to 1/2 and 1/4. One can see that the more frames with different poses and deformations are seen during training, the better the reconstruction quality is. This is expected since a larger number of frames may better sample the possible space of poses and deformations. 3) In the third group of Table 5.3, the *AMVIoU* is evaluated on the template mesh animated with the results of *PoseNet*, which is referred to as *PoseNet-only*. One can see that on average, the *AMVIoU* is improved by around 4%. Since most non-rigid deformations rather happen locally, the difference is visually even more significant, as shown in Figure 5.11. Especially, the skirt is correctly deformed according to the input image, whereas

AMVIoU, RVIoU, and SVIoU (in %) on S1 sequence			
Method	**AMVIoU↑**	**RVIoU↑**	**SVIoU↑**
HMR (Kanazawa et al., 2018)	62.25	61.7	68.85
HMMR (Kanazawa et al., 2019)	65.98	65.58	70.77
LiveCap (Habermann et al., 2019)	56.02	54.21	77.75
DeepHuman (Zheng et al., 2019)	-	-	**91.57**
Ours	**87.2**	**87.03**	89.26
MVBL	91.74	91.72	92.02

AMVIoU, RVIoU and SVIoU (in %) on S2			
Method	**AMVIoU↑**	**RVIoU↑**	**SVIoU↑**
HMR (Kanazawa et al., 2018)	59.79	59.1	66.78
HMMR (Kanazawa et al., 2019)	62.64	62.03	68.77
LiveCap (Habermann et al., 2019)	60.52	58.82	77.75
DeepHuman (Zheng et al., 2019)	-	-	**91.57**
Ours	**83.73**	**83.49**	89.26
MVBL	89.62	89.67	92.02

AMVIoU, RVIoU and SVIoU (in %) on S3			
Method	**AMVIoU↑**	**RVIoU↑**	**SVIoU↑**
HMR (Kanazawa et al., 2018)	59.05	58.73	63.12
HMMR (Kanazawa et al., 2019)	61.73	61.32	67.14
LiveCap (Habermann et al., 2019)	61.55	60.47	75.6
DeepHuman (Zheng et al., 2019)	-	-	79.66
Ours	**85.75**	**85.55**	**88.27**
MVBL	90.31	90.21	91.53

AMVIoU, RVIoU, and SVIoU (in %) on S4 sequence			
Method	**AMVIoU↑**	**RVIoU↑**	**SVIoU↑**
HMR (Kanazawa et al., 2018)	65.1	64.66	70.84
HMMR (Kanazawa et al., 2019)	63.79	63.29	70.23
LiveCap (Habermann et al., 2019)	59.96	59.02	72.16
DeepHuman (Zheng et al., 2019)	-	-	84.15
Ours	**82.53**	**82.22**	**86.66**
MVBL	88.14	88.03	89.66

Table 5.2: Surface deformation accuracy. Note that the proposed method again outperforms all other monocular methods and is close to the multi-view baseline. Further note that for (Zheng et al., 2019) an evaluation of the multi-view IoU is not possible since their output is always in local image space that cannot be brought to global space.

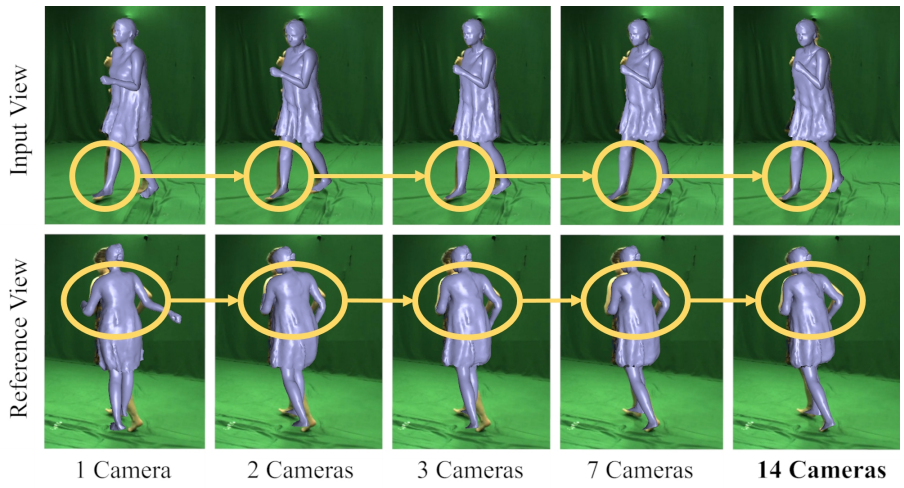


Figure 5.8: Ablation for the number of *cameras* used during training. The most significant improvement happens when adding one additional camera to the monocular setting. But also adding further cameras consistently improves the result, as the yellow circles indicate.

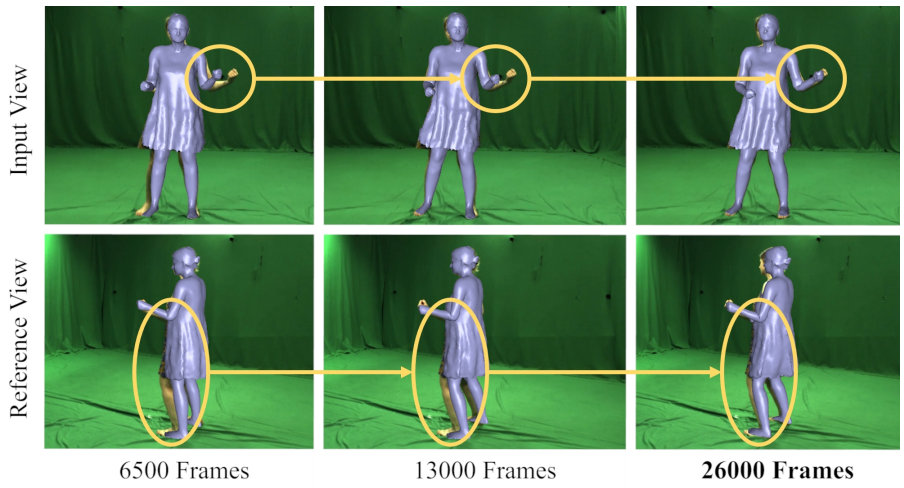


Figure 5.9: Ablation for the number of *frames* used during training. The more frames are used during training, the better the result becomes as the network can better sample the possible pose and deformation space.

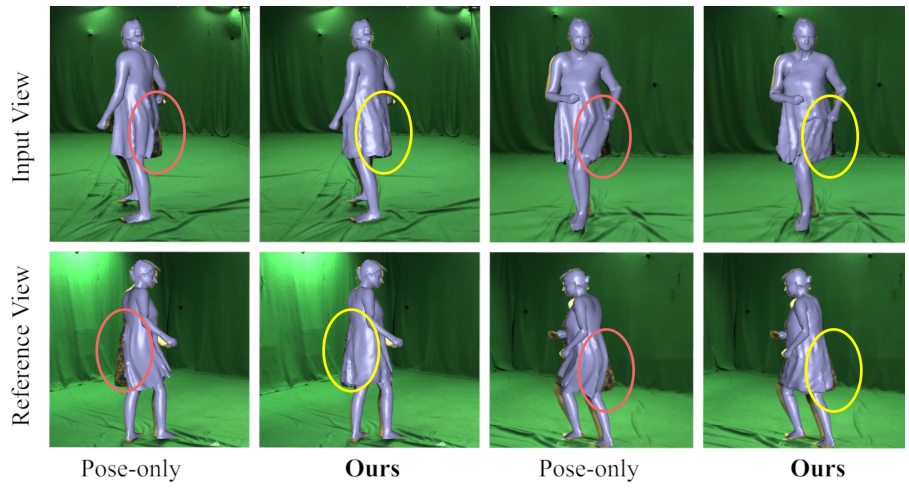


Figure 5.10: DeepCap result from the input view and a reference view that was not used for tracking. Note that the *DefNet* can even regress deformations along the camera viewing axis of the input camera (second column), and it can correctly deform surface parts that are occluded (fourth column).

the *PoseNet-only* result cannot fit the input due to the limitation of skinning. Figure 5.10 shows the *PoseNet-only* result and the final result on one of the evaluation sequences where a reference view is available. The deformed template also looks plausible from a reference view that was not used for tracking. Importantly, *DefNet* can correctly regress deformations that are along the camera viewing direction of the input camera (see reference view in the second column) and surface parts that are even occluded (see reference view in the fourth column). This implies that the weak multi-view supervision during training lets the network learn the entire 3D surface deformation of the human body. 4) finally, in Figure 5.12, the impact of the domain adaptation step is visually demonstrated. It becomes obvious that the refinement drastically improves the pose as well as the non-rigid deformations so that the input can be matched at much higher accuracy. Further, no additional input is required for the refinement as the losses can be directly adapted to the monocular setting.

5.8.6 *Applications*

The presented method enables driving 3D characters just from a monocular RGB video (see Figure 5.4). As the only device needed is a single color camera, DeepCap can be easily used in daily life scenarios. Further, as it is also accounted for non-rigid surface deformations, the proposed method also enhances the realism of the virtual characters. The proposed approach also allows augmenting a video as shown in Figure 5.13. Since the entire 3D geometry is tracked, the augmented

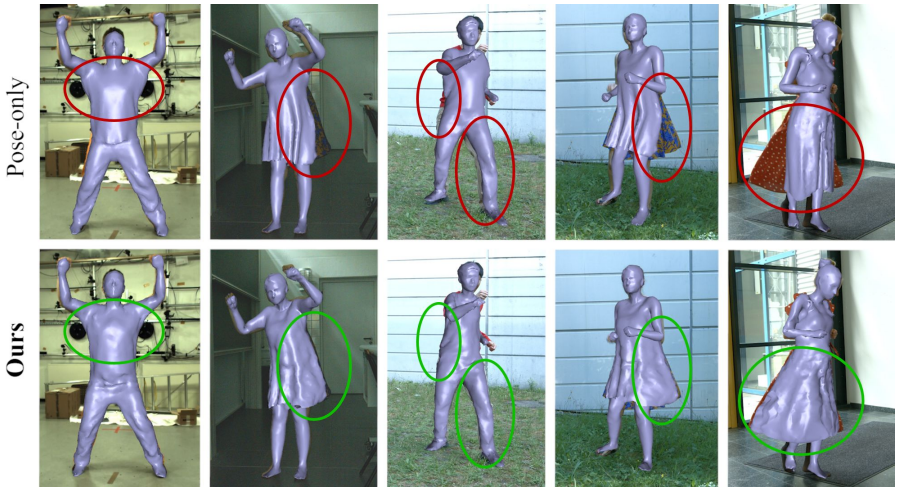


Figure 5.11: *PoseNet + DefNet* vs. *PoseNet-only*. *DefNet* can deform the template to accurately match the input, especially for loose clothing. In addition, *DefNet* also corrects slight errors in the pose and typical skinning artifacts.

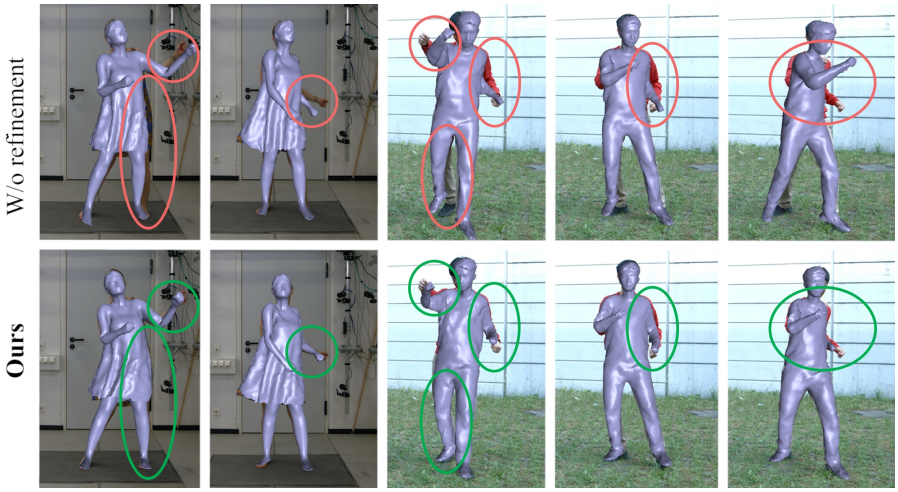


Figure 5.12: Impact of the in-the-wild domain adaption step. Note that after the network refinement, both the pose as well as the deformations better match the input.

3DPCK and AMVIoU (in %) on S4 sequence		
Method	**3DPCK↑**	**AMVIoU↑**
1 camera view	62.11	65.11
2 camera views	93.52	78.44
3 camera views	94.70	79.75
7 camera views	95.95	81.73
6500 frames	85.19	73.41
13000 frames	92.25	78.97
PoseNet-only	96.74	78.51
Ours(14 views, 26000 frames)	**96.74**	**82.53**

Table 5.3: Ablation study. The number of cameras and the number of frames used during training is evaluated in terms of the *3DPCK* and *AMVIoU* metrics. Adding more cameras and frames consistently improves the quality of reconstruction. Further, *DefNet* improves the *AMVIoU* compared to pure pose estimation.

texture is also aware of occlusions in contrast to pure image-based augmentation techniques.

5.9 LIMITATIONS AND FUTURE WORK

Conceptually, both representations, pose and non-rigid deformations, are decoupled. Nevertheless, since the predicted poses during training are not perfect, the *DefNet* also deforms the graph to account for wrong poses to a certain degree. The proposed method was also tested on subjects that were not used for training but who wear the same clothing as the training subject. Although the presented method still performs reasonable, the overall accuracy drops as the subject's appearance was never observed during training. Further, DeepCap can fail for extreme poses, e.g., a handstand, that were not observed during training.

5.10 CONCLUSION

A learning-based approach for monocular dense human performance capture using only weak multi-view supervision was presented. In contrast to existing methods, the proposed approach directly regresses poses and surface deformations from neural networks, produces temporal surface correspondences, preserves the skeletal structure of the human body, and can handle loose clothes. Qualitative and quantitative results in different scenarios show that DeepCap produces a more accurate 3D reconstruction of pose and non-rigid deformation than existing methods. Incorporating the hands and the face into the mesh

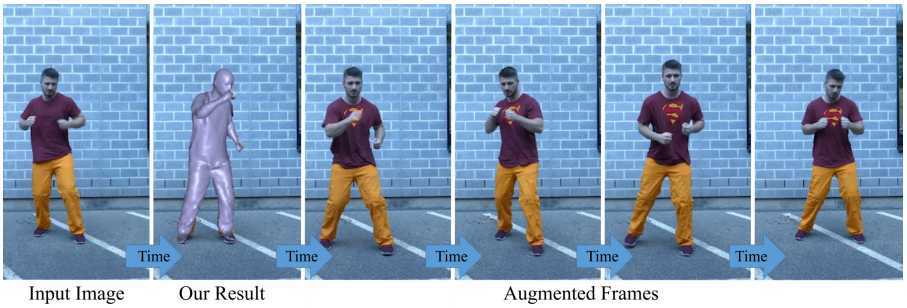


Figure 5.13: Video augmentation. The proposed method can be used to augment a video with textures like the logo on the T-shirt. Since the underlying 3D geometry is tracked, the proposed method also accounts for occlusions of the augmented texture.

representation could be an interesting direction for future research, which would enable joint tracking of the body, facial expressions, and hand gestures. Despite that, it could also be interesting to incorporate physically more correct multi-layered representations to model the garments even more realistically.

The last two chapters presented new methods for faster and more accurate monocular human performance capture compared to the previous state of the art. While these approaches focused mainly on capturing the human in the image, it is equally important to be able to also synthesize photo-real humans. Thus, in the following chapter, a novel learning-based approach is presented, which creates animatable and photorealistic 3D characters that can be rendered in real time where the motion can be completely controlled by a user. Importantly, these characters can be obtained solely from 2D multi-view video footage, and no 3D ground truth is required.

6

REAL-TIME DEEP DYNAMIC CHARACTERS

While the previous chapters of the thesis focused on capturing the human performance, this chapter proposes a deep video realistic 3D human character model (published as Habermann et al., 2021a) displaying highly realistic shape, motion, and dynamic appearance learned in a new weakly supervised way from multi-view imagery. In contrast to previous work, the proposed controllable 3D character displays dynamics, e.g., the swing of the skirt, dependent on skeletal body motion in an efficient data-driven way, without requiring complex physics simulation. The character model also features a learned dynamic texture model that accounts for photo-realistic motion-dependent appearance details, as well as view-dependent lighting effects. During training, the difficult dynamic 3D capture of the entire human is not required; instead, the model can be entirely trained on multi-view video in a weakly supervised manner. To this end, a parametric and differentiable character representation is proposed, which allows the approach to model coarse and fine dynamic deformations, e.g., garment wrinkles, as explicit space-time coherent mesh geometry that is augmented with high-quality dynamic textures dependent on motion and viewpoint. As input to the model at test time, only an arbitrary 3D skeleton motion is required, making it directly compatible with the established 3D animation pipeline in Computer Graphics. A novel graph convolutional network architecture is introduced, which enables motion-dependent deformation learning of body and clothing, including dynamics, and a neural generative dynamic texture model creates corresponding dynamic texture maps. It is shown that by merely providing new skeletal motions, the model creates motion-dependent surface deformations, physically plausible dynamic clothing deformations, as well as video-realistic surface textures at a much higher level of detail than the previous state of the art approaches, and even in real time.

6.1 INTRODUCTION

Animatable and photo-realistic virtual 3D characters are of enormous importance nowadays. With the rise of computer graphics in movies, games, telepresence, and many other areas, 3D virtual characters are everywhere. Recent developments in virtual and augmented reality and the resulting immersive experience further boosted the need for virtual characters as they now become part of our real lives. However, generating realistic characters still requires manual intervention, expensive equipment, and the resulting characters are either difficult to

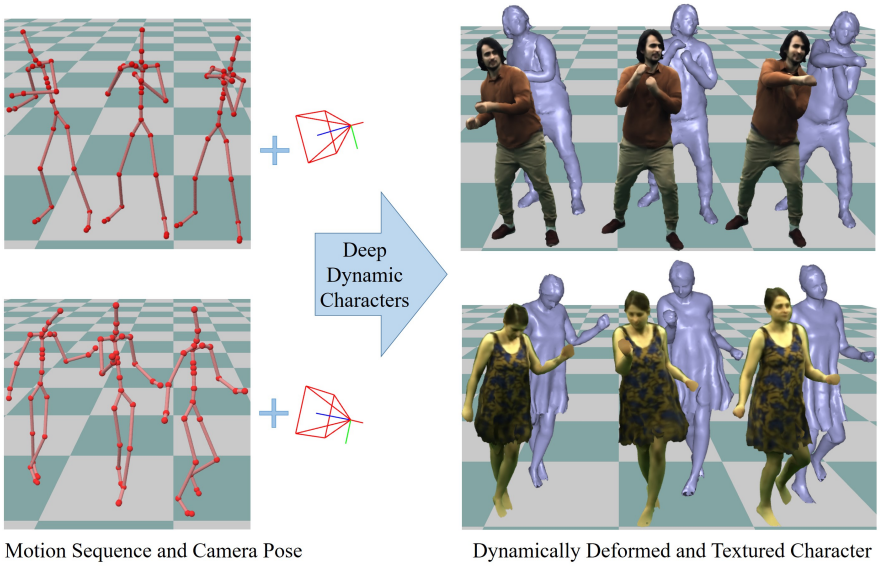


Figure 6.1: The proposed learning-based method takes a sequence of poses and regresses the motion-dependent dynamic surface deformation of a person-specific template. To further enhance realism, the approach also predicts a motion- and view-dependent dynamic texture map. Note that the final textured model looks video realistic and can be used in many applications, e.g., neural video synthesis or interactive character editing.

control or not realistic. Therefore, the goal is to learn digital characters which are both realistic and easy to control and can be learned directly from a multi-view video.

It is a complicated process to synthesize realistic-looking images of deforming characters following the conventional computer graphics pipeline. The static geometry of real humans is typically represented with a mesh obtained with 3D scanners. In order to pose or animate the mesh, a skeleton has to be attached to the geometry, i.e., rigging and skinning techniques can then be used to deform the mesh according to the skeletal motion. While these approaches are easy to control and efficient, they lack realism as the non-rigid deformations of clothing are not modeled, e.g., the swinging of a skirt. While physics simulation can address this, it requires expert knowledge as it is hard to control. Further, these techniques are either computationally expensive or not robust to very articulated poses leading to glitches in the geometry. Finally, expensive physically based rendering techniques are needed to render realistic images of the 3D character. Those techniques are not only time-consuming but also require expert knowledge and manual parameter tuning.

To model clothing deformations, recent work combines classical skinning with a learned mapping from skeletal motion to non-rigid deformations and learns the model from data. One line of work learns from real data, but the results either lack realism (Ma et al., 2020) or

are limited to partial clothing, e.g., a T-shirt (Lähner et al., 2018). More importantly, as they rely on ground truth registered 3D geometry, they require expensive 3D scanners and challenging template registration. Another line of work tries to learn from a large database of simulated clothing (Guan et al., 2012; Patel et al., 2020). While they can generalize across clothing categories and achieve faster run-times than physics simulations, the realism is still limited by the physics engine used for training data generation.

Furthermore, the texture dynamics are not captured by the aforementioned methods, although they are crucial to achieve photo-realism. Monocular neural rendering approaches (Chan et al., 2019; Liu et al., 2020b, 2019a) for humans learn a mapping from a CG rendering to a photo-realistic image, but their results have limited resolution and quality and struggle with consistency when changing pose and viewpoint. The most related works (Casas et al., 2014; Shysheya et al., 2019; Xu et al., 2011) are the ones leveraging multi-view imagery for creating animatable characters. However, all of them are not modeling a motion-dependent deforming geometry and/or view-dependent appearance changes.

To overcome the limitations of traditional skinning, the requirement of direct 3D supervision of recent learning-based methods, as well as their lack of dynamic textures, a learning-based method is proposed that predicts the non-rigid character surface deformation of the full human body as well as a dynamic texture from skeletal motion *using only weak supervision in the form of multi-view images during training*. At the core of the method is a differentiable character (with neural networks parameterizing dynamic textures and non-rigid deformations) which can generate images differentiable with respect to its parameters. This allows one to train directly with multi-view image supervision using analysis by synthesis and back-propagation, instead of pre-computing 3D mesh registrations, which is difficult, tedious, and prone to error. In designing differentiable characters, the key insight is to learn as much of the deformation as possible in geometry space and produce the subtle fine details in texture space. Compared to learning geometry deformations in image space, this results in much more coherent results when changing viewpoint and pose. To this end, a novel graph convolutional network architecture is proposed which takes a temporal motion encoding and predicts the surface deformation in a coarse to fine manner using the new fully differentiable character representation. The learned non-rigid deformation and dynamic texture not only account for dynamic clothing effects such as the swinging of a skirt caused by the actor's motion or fine wrinkles appearing in the texture but also fixes traditional skinning artifacts such as candy wrappers. Moreover, as the dynamic texture is conditioned on the camera pose, the proposed approach can also model view-dependent effects, e.g., specular surface reflections.

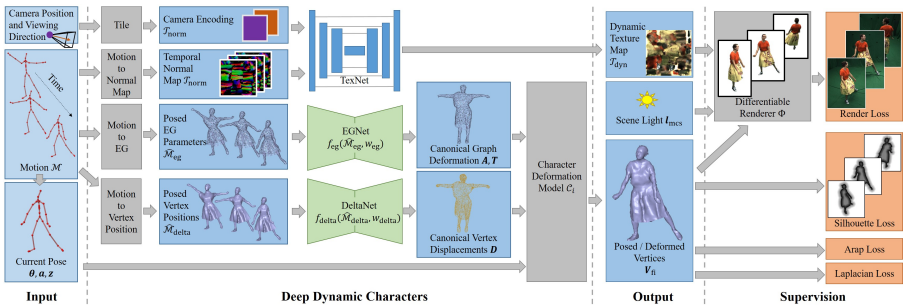


Figure 6.2: Overview. The proposed method takes a motion sequence as input. The pose information is converted into task-specific representations making the regression task easier for the network as input and output share the same representation. Then the two networks regress the motion-dependent coarse and fine deformations in the canonical pose. Given motion and deformations, the deformation layer outputs the posed and deformed character. Further, the TexNet regresses a motion- and view-dependent dynamic texture map. The regressed geometry, as well as texture, are weakly supervised based on multi-view 2D images.

In summary, the contributions are:

- The first learning-based real-time approach that takes a skeletal motion and camera pose as input and predicts the motion-dependent surface deformation and motion- and view-dependent texture for the full human body using direct image supervision.
- A differentiable 3D character representation which can be trained from coarse to fine (Section 6.3).
- A graph convolutional architecture allowing one to formulate the learning problem as a graph-to-graph translation (Section 6.5).
- A new benchmark dataset, called *DynaCap*, containing 5 actors captured with a dense multi-view system which will be made publicly available (Section 6.9.1).

The resulting dynamic characters can be driven either by motion capture approaches or by interactive editing of the underlying skeleton. This enables many exciting applications in gaming and movies, such as more realistic character control as the character deformation and texture will account for dynamic effects. Qualitative and quantitative results show that the proposed approach is a significant step forward towards photo-realistic and animatable full-body human avatars.

6.2 OVERVIEW

Given multi-view images for training, the goal is to learn a poseable 3D character with dense deforming geometry of the full body and view- and motion-dependent textures that can be driven just by posing

a skeleton and defining a camera view. A weakly supervised learning method with only multi-view 2D supervision is proposed in order to remove the need for detailed 3D ground truth geometry and 3D annotations. Once trained, the network takes the current pose and a frame window of past motions of a moving person as input and then outputs the motion-dependent geometry and texture, as shown in Figure 6.2. Note that the deformed geometry captures not only per-bone rigid transformations via classical skinning but also non-rigid deformations of clothing dependent on the current pose as well as the velocity and acceleration derived from the past motions. In the following, a novel deformable character model (Section 6.3) is proposed, and the data acquisition process is described (Section 6.4). To regress the non-rigid deformations, a coarse-to-fine approach is proposed. First, the deformation is regressed as rotations and translations of a coarse embedded graph (Section 6.5) only using multi-view foreground images as supervision signal. As a result, a posed and deformed character can be obtained that already matches the silhouettes of the multi-view images. Next, a differentiable rendering layer is defined, which allows the approach to optimize the scene lighting, which accounts for white balance shift and directional light changes (Section 6.6). Finally, the second network regresses per-vertex displacements to account for finer wrinkles and deformations that cannot be captured by the embedded deformation. This layer can be trained using the foreground masks again, but in addition, it is also supervised with a dense rendering loss using the previously optimized scene lighting (Section 6.7). Last, the dynamic texture network takes a view and motion encoding in texture space and outputs a dynamic texture (Section 6.8) to further enhance the realism of the 3D character. Similar to before, the texture network is weakly supervised using the differentiable renderer. Note that none of the proposed components requires ground truth 3D geometry and can be entirely trained weakly supervised.

6.3 CHARACTER DEFORMATION MODEL

Next, the data acquisition for the template is described as well as the skeleton, embedded graph, and vertex displacement representations which are then combined in the final character representation.

6.3.1 *Template Acquisition*

The proposed method is person-specific and requires a 3D template model of the actor. First, the actor is scanned in T-pose using a 3D scanner (*Treedys* 2020). Next, a commercial multi-view stereo reconstruction software (*PhotoScan* 2016) is used to reconstruct the 3D mesh with a static texture $\mathcal{T}_{st}$ and the reconstructed mesh is downsampled

to a resolution of around 5000 vertices. Like (Habermann et al., 2020, 2019), the mesh is manually segmented using the common human parsing labels, and per-vertex rigidity weights s_i are defined to model different degrees of deformation for different materials, where low rigidity weights allow more non-rigid deformations and vice versa, e.g., the skin has higher rigidity weights than clothing.

6.3.2 *Skeleton*

The template mesh is manually rigged to a skeleton. Here, the skeleton is parameterized as the set $\mathcal{S} = \{\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{z}\}$ with joint angles $\boldsymbol{\theta} \in \mathbb{R}^{57}$, global rotation $\boldsymbol{\alpha} \in \mathbb{R}^3$, and global translation $\mathbf{z} \in \mathbb{R}^3$, where skinning weights are automatically computed using Blender (*Blender* 2020). This allows the deformation of the mesh for a given pose by using dual quaternion skinning (Kavan et al., 2007).

6.3.3 *Embedded Deformation*

As discussed before, the traditional skinning process alone is hardly able to model non-rigid deformations such as the swinging of a skirt. To address this issue, the non-rigid deformations are modeled in the canonical pose from coarse to fine *before* applying dual quaternion skinning. On the coarse level, large deformations are captured with the embedded deformation representation (Sorkine and Alexa, 2007; Sumner et al., 2007), which requires a small number of parameters. An embedded graph $\mathcal{G}$ is constructed consisting of K nodes (K is around 500 in the experiments) by downsampling the mesh. The embedded graph $\mathcal{G}$ is parameterized with $\mathbf{A} \in \mathbb{R}^{K\times 3}$ and $\mathbf{T} \in \mathbb{R}^{K\times 3}$, where each row k of $\mathbf{A}$ and $\mathbf{T}$ is the local rotation $\mathbf{a}_k \in \mathbb{R}^3$ in the form of Euler angles and local translation $\mathbf{t}_k \in \mathbb{R}^3$ of node k with respect to the initial position $\mathbf{g}_k$ of node k. The connectivity of the graph node k can be derived from the connectivity of the downsampled mesh and is denoted as $\mathcal{N}_{\mathrm{n}}(k)$. To deform the original mesh with the embedded graph, the movement of each vertex on the original mesh is calculated as a linear combination of the movements of all the nodes of the embedded graph. Here, the weights $w_{i,k} \in \mathbb{R}$ for vertex i and node k are computed based on the geodesic distance between the vertex i and the vertex on the original mesh that has the smallest Euclidean distance to the node k, where the weight is set to zero if the geodesic distance exceeds a certain threshold. The set of nodes that finally influences the movement of the vertex i is denoted as $\mathcal{N}_{\mathrm{vn}}(i)$.

6.3.4 *Vertex Displacements*

On the fine level, in addition to the embedded graph, which models large deformations, vertex displacements are used to recover fine-scale deformations, where a displacement $\mathbf{d}_i \in \mathbb{R}^3$ is assigned to each vertex i. Although regressing so many parameters is not an easy task, the training of the vertex displacement can still be achieved since the embedded graph captures most deformations on a coarse level. Thus, the regressed displacements, the network has to learn, are rather small.

6.3.5 *Character Deformation Model*

Given the skeletal pose $\boldsymbol{\theta}$, $\boldsymbol{\alpha}$, $\mathbf{z}$, the embedded graph parameters $\mathbf{A}$, $\mathbf{T}$, and the vertex displacements $\mathbf{d}_i$, each vertex i can be deformed with the function

$$C_i(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{z}, \mathbf{A}, \mathbf{T}, \mathbf{d}_i) = \mathbf{v}_i \tag{6.1}$$

which defines the final character representation. Specifically, first, the embedded deformation and the per-vertex displacements are applied to the template mesh in canonical pose, which significantly simplifies the learning of non-rigid deformations by alleviating ambiguities in the movements of mesh vertices caused by pose variations. Thus, the deformed vertex position is given as

$$\mathbf{y}_i = \mathbf{d}_i + \sum_{k \in \mathcal{N}_{\text{vn}}(i)} w_{i,k}(R(\mathbf{a}_k)(\hat{\mathbf{v}}_i - \mathbf{g}_k) + \mathbf{g}_k + \mathbf{t}_k), \tag{6.2}$$

where $\hat{\mathbf{v}}_i$ is the initial position of vertex i in the template mesh. $R : \mathbb{R}^3 \rightarrow SO(3)$ converts the Euler angles to a rotation matrix. The skeletal pose is applied to the deformed vertex $\mathbf{y}_i$ in canonical pose to obtain the deformed and posed vertex in the global space

$$\mathbf{v}_i = \mathbf{z} + \sum_{k \in \mathcal{N}_{\text{vn}}(i)} w_{i,k}(R_{\text{sk},k}(\boldsymbol{\theta}, \boldsymbol{\alpha})\mathbf{y}_i + t_{\text{sk},k}(\boldsymbol{\theta}, \boldsymbol{\alpha})), \tag{6.3}$$

where the rotation $R_{\text{sk},k}$ and the translation $t_{\text{sk},k}$ are derived from the skeletal pose using dual quaternion skinning, and $\mathbf{z}$ is the global translation of the skeleton. Note that Equation 6.2 and 6.3 are fully differentiable with respect to pose, embedded graph, and vertex displacements. Thus, gradients can be propagated in learning frameworks. The final model does not only allow the approach to pose the mesh via skinning but also to model non-rigid surface deformations in a coarse to fine manner via embedded deformation and vertex displacements. Further, it disentangles the pose and the surface deformation, where the latter is represented in the canonical pose space.

The main difference to data-driven body models, e.g., SMPL (Loper et al., 2015), is that the character formulation allows posing, deforming, and texturing using an effective and simple equation that is differentiable to all its input parameters. SMPL and other human body models

do not account for deformations, e.g., clothing, and they also do not provide a texture. The specific formulation allows seamless integration into a learning framework and learning its parameters conditioned on skeletal motion (and camera pose) as well as adding spatial regularization from coarse to fine, which is important when considering a weakly supervised setup.

6.4 DATA CAPTURE AND MOTION PREPROCESSING

For supervision, the proposed method requires multi-view images and foreground masks of the actor performing a wide range of motions at varying speeds to sample different kinds of dynamic deformations of clothing caused by the body motion. Thus, the subject is placed in a multi-camera capture studio with a green screen, and a sequence with $C = 120$ synchronized and calibrated 4K cameras is recorded at 25 frames per second. Color keying is applied to segment the foreground, and then the foreground masks are converted into distance transform images (Borgefors, 1986). The fth frame of camera c and its corresponding distance transform image and the foreground mask are denoted as $\mathcal{I}_{c,f}$, $\mathcal{D}_{c,f}$, and $\mathcal{F}_{c,f}$, respectively.

Further, the human motions are tracked using a multi-view markerless motion capture system (*The Captury* 2020). The tracked motion of the fth frame is denoted as $\mathcal{S}_f = \{\boldsymbol{\theta}_f, \boldsymbol{\alpha}_f, \mathbf{z}_f\}$. Next, the temporal window of motions $\mathcal{M}_t = \{\mathcal{S}_f : f \in \{t-F, ..., t\}\}$ is normalized for geometry and texture generation separately as it is very hard to sample all combinations of rigid transforms and joint angle configurations during training data acquisition. The normalization for geometry generation is based on two observations: 1) The global position of the motion sequence should not influence the dynamics of the geometry; therefore the global translation of $\mathcal{S}_f$ is normalized across different temporal windows of motions while keeping relative translations between the frames in each temporal window, i.e., the translation is set to $\hat{\mathbf{z}}_t = \mathbf{0}$ and $\hat{\mathbf{z}}_{t'} = \mathbf{z}_{t'} - \mathbf{z}_t$ for $t' \in \{t-F, ..., t-1\}$ where $\mathbf{0}$ is the zero vector in $\mathbb{R}^3$. 2) The rotation around the y axis will not affect the geometry generation as it is in parallel with the gravity direction; thus, similar to normalizing the global translation, the rotation is set to $\hat{\boldsymbol{\alpha}}_{y,t} = 0$ and $\hat{\boldsymbol{\alpha}}_{y,t'} = \boldsymbol{\alpha}_{y,t'} - \boldsymbol{\alpha}_{y,t}$. The temporal window of the normalized motions for geometry generation is denoted as $\hat{\mathcal{M}}_t = \{\hat{\mathcal{S}}_f : f \in \{t-F, ..., t\}\}$, where $\hat{\mathcal{S}}_f = \{\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\alpha}}_f, \hat{\mathbf{z}}_f\}$. For texture generation, only the global translation is normalized, but not the rotation around the y axis to get the normalized motions $\tilde{\mathcal{M}}_t$, since the goal is to generate view-dependent textures where the subject's relative direction towards the light source and therefore the rotation around the y axis matters. In all the results, the frame window is set to $F = 2$. For readability reasons, it is assumed that t is fixed and the subscript is dropped.

6.5 EMBEDDED DEFORMATION REGRESSION

Next, it is described how the coarse embedded deformations of the character model are regressed given a skeletal motion as input. To this end, a novel graph convolutional architecture is proposed, and it is described how the network is supervised.

6.5.1 *Embedded Deformation Regression*

With the skeletal pose alone, the non-rigid deformations of the skin and clothes cannot be generated. Therefore, an embedded graph network, *EGNet*, is introduced to produce coarse-level deformations. *EGNet* learns a mapping from the temporal window of normalized motions $\hat{\mathcal{M}}$ to the rotations $\mathbf{A}$ and translations $\mathbf{T}$ of the embedded graph defined in the canonical pose for the current frame (i.e., the last frame of the temporal window). *EGNet* learns deformations correlated to the velocity and acceleration at the current frame since it takes the pose of the current frame as well as the previous two frames as input. Directly regressing $\mathbf{A}$ and $\mathbf{T}$ from the normalized skeletal motion $\hat{\mathcal{M}}$ is challenging as the input and output are parameterized in a different way, i.e., $\hat{\mathcal{M}}$ represents skeleton joint angles while $\mathbf{A}$ and $\mathbf{T}$ model rotation and translation of the graph nodes. To address this issue, this regression task is formulated as a graph-to-graph translation problem rather than a skeleton-to-graph one. Specifically, the embedded graph is posed with the normalized skeletal motion $\hat{\mathcal{M}}$ using dual quaternion skinning (Kavan et al., 2007) to obtain the rotation and translation parameters $\hat{\mathcal{M}}_{\text{eg}} \in \mathbb{R}^{K\times 6(F+1)}$ of the embedded graph. Therefore, the mapping of *EGNet* can be formulated as $f_{\text{eg}}(\hat{\mathcal{M}}_{\text{eg}}, \mathbf{w}_{\text{eg}}) = (\mathbf{A}, \mathbf{T})$, which takes the posed embedded graph rotations and translations $\hat{\mathcal{M}}_{\text{eg}}$ and learnable network weights $\mathbf{w}_{\text{eg}}$ as inputs and outputs the embedded deformation $(\mathbf{A}, \mathbf{T})$ in canonical pose. Using the character representation defined in Equation 6.1, the posed and coarsely deformed character is defined as

$$\mathcal{C}_i(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{z}, f_{\text{eg}}(\hat{\mathcal{M}}_{\text{eg}}, \mathbf{w}_{\text{eg}}), \mathbf{0}) = \mathbf{v}_{\text{co},i}. \tag{6.4}$$

Here, $\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{z}$ are the unnormalized pose of the last frame of the motion sequence, and the displacements are set to zero. Next, the novel graph convolutional architecture of EGNet is explained.

6.5.2 *Structure-aware Graph Convolution*

Importantly, the graph is fixed as the proposed method is person-specific. Thus, the spatial relationship between the graph nodes and their position implicitly encodes a strong prior. For example, a node that is mostly attached to skin vertices will deform very different

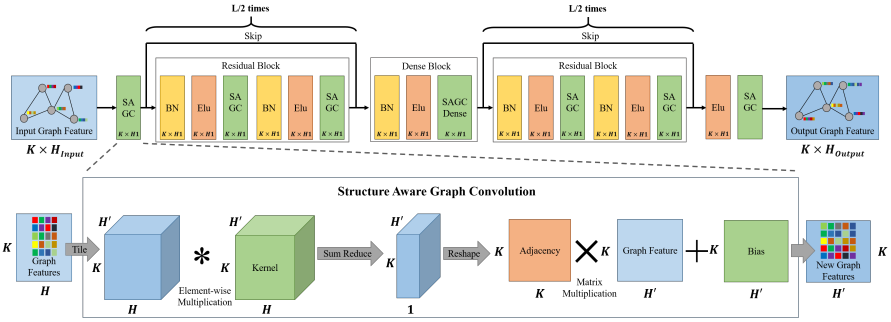


Figure 6.3: Structure aware graph convolutional network (top) as well as a detailed illustration of the proposed Structure Aware Graph Convolution (bottom).

than nodes that are mainly connected to vertices of a skirt region. This implies that learnable node features require different properties depending on which node is considered. However, recent graph convolutional operators (Defferrard et al., 2016) apply the same filter on every node, which contradicts the above requirements. Therefore, the proposed approach aims for a graph convolutional operator that applies an individual kernel per node.

Thus, a new *Structure-aware Graph Convolution (SAGC)* is introduced. To define the per-node SAGC, it is assumed that an input node feature $\mathbf{f}_k \in \mathbb{R}^H$ of size H is given, and the output feature dimension is H' for a node k. Now, the output feature $\mathbf{f}'_k$ can be computed as

$$\mathbf{f}'_k = \mathbf{b}_k + \sum_{l \in \mathcal{N}_R(k)} a_{k,l} \mathbf{K}_l \mathbf{f}_l \tag{6.5}$$

where $\mathcal{N}_R(k)$ is the R-ring neighbourhood of the graph node k. $\mathbf{b}_k \in \mathbb{R}^{H'}$ and $\mathbf{K}_l \in \mathbb{R}^{H' \times H}$ are a trainable bias vector and kernel matrix. $a_{k,l}$ is a scalar weight that is computed as

$$a_{k,l} = \frac{r_{k,l}}{\sum_{l \in \mathcal{N}_R(k)} r_{k,l}} \tag{6.6}$$

where for a node k $r_{k,l}$ is the inverse ring value, e.g., for the case $l = k$ the value is R and for the direct neighbours of k the value is $R - 1$. More intuitively, the operator computes a linear combination of modified features $\mathbf{K}_l \mathbf{f}_l$ of node k and neigbouring nodes l within the R-ring neighbourhood weighted by $a_{k,l}$ that has a linear falloff to obtain the new feature for node k. Importantly, each node has its own learnable kernel $\mathbf{K}_l$ and bias $\mathbf{b}_k$ weights allowing features at different locations in the graph to account for different spatial properties. As shown at the bottom of Figure 6.3, the features for each node can be efficiently computed in parallel, and by combining all the per node input/output features, one obtains the corresponding input/output feature matrices $\mathbf{F}_k, \mathbf{F}'_k$.

6.5.3 *Structure-aware Graph Convolutional Network*

The structure-aware graph convolutional network (SAGCN) takes as input a graph feature matrix $\mathbf{F}_{\text{input}} \in \mathbb{R}^{K \times H_{\text{input}}}$ and outputs a new graph feature matrix $\mathbf{F}_{\text{output}} \in \mathbb{R}^{K \times H_{\text{output}}}$ (see Figure 6.3). First, $\mathbf{F}_{\text{input}}$ is convolved with the SAGC operator, resulting in a feature matrix of size $K \times H_1$. Inspired by the ResNet architecture (He et al., 2016), also so-called residual blocks are used that take the feature matrix of size $K \times H_1$ and output a feature matrix of the same size. Input and output feature matrices are connected via skip connections which prevent vanishing gradients, even for very deep architectures. A residual block consists of two chains of a batch normalization, an Elu activation function, and a SAGC operation. For a very large number of graph nodes, the local features can barely spread through the entire graph. To still allow the network to share features between far nodes, a so-called dense block is proposed, which consists of a batch normalization, an Elu activation, and a SAGC operator. Importantly, for this specific dense block, all entries of the weighting matrix are set to $a_{k,l} = 1$ which allows the network to share features between far nodes. In total, L residual blocks are used, half of them before and half of them after the dense block. The last layers (Elu and SAGC) resize the features to the desired output size.

Now, EGNet can be defined as a SAGCN architecture where the graph is given as the embedded graph $\mathcal{G}$. The input feature matrix is given by the normalized embedded graph rotations and translations $\hat{\mathcal{M}}_{\text{eg}}$ and the output is the deformation parameters $(\mathbf{A}, \mathbf{T})$. Thus, the input and output feature sizes are $H_{\text{input}} = 6(F+1)$ and $H_{\text{output}} = 6$, respectively. Further, the hyperparameters are set to $H_1 = 16$, $L = 8$, and $R = 3$. As $\mathcal{G}$ only contains around 500 nodes, a dense block is not employed at this stage.

6.5.4 *Weakly Supervised Losses*

To train the weights $\mathbf{w}_{\text{eg}}$ of EGNet $f_{\text{eg}}(\hat{\mathcal{M}}_{\text{eg}}, \mathbf{w}_{\text{eg}})$, only a weakly supervised loss is imposed on the posed and deformed vertices $\mathbf{V}_{\text{co}}$ and on the regressed embedded deformation parameters $(\mathbf{A}, \mathbf{T})$ directly as

$$\mathcal{L}_{\text{eg}}(\mathbf{V}_{\text{co}}, \mathbf{A}, \mathbf{T}) = \mathcal{L}_{\text{sil}}(\mathbf{V}_{\text{co}}) + \mathcal{L}_{\text{arap}}(\mathbf{A}, \mathbf{T}). \tag{6.7}$$

Here, the first term is a multi-view image-based data loss, and the second term is a spatial regularizer.

6.5.4.1 *Silhouette Loss*

The multi-view silhouette loss

$$\begin{aligned}\mathcal{L}_{\text{sil}}(\mathbf{V}_{\text{co}}) = &\sum_{c=1}^{C}\sum_{i\in\mathcal{B}_c}\rho_{c,i}\|\mathcal{D}_c\left(\pi_c\left(\mathbf{v}_{\text{co},i}\right)\right)\|^2 \\ &+\sum_{c=1}^{C}\sum_{\mathbf{p}\in\{\mathbf{u}\in\mathbb{R}^2|\mathcal{D}_c(\mathbf{u})=0\}}\|\pi_c\left(\mathbf{v}_{\text{co},\mathbf{p}}\right)-\mathbf{p}\|^2\end{aligned} \tag{6.8}$$

ensures that the silhouette of the projected character model aligns with the multi-view image silhouettes in an analysis-by-synthesis manner. Therefore, a bi-sided loss is employed. The first part of Equation 6.8 is a model-to-data loss which enforces that the projected boundary vertices are pushed to the zero contour line of the distance transform image $\mathcal{D}_c$ for all cameras c. Here, π_c is the perspective camera projection of camera c and $\rho_{c,i}$ is a scalar weight accounting for matching image and model normals (Habermann et al., 2019). $\mathcal{B}_c$ is the set of boundary vertices, e.g., the vertices that lie on the boundary after projecting onto camera view c. $\mathcal{B}_c$ can be efficiently computed using the differentiable renderer, which is introduced later, by rendering out the depth maps and checking if a projected vertex lies near a background pixel in the depth map. The second part of Equation 6.8 is a data-to-model loss which ensures that all silhouette pixels $\{\mathbf{u}\in\mathbb{R}^2|\mathcal{D}_c(\mathbf{u})=0\}$ are covered by their closest vertex $\mathbf{v}_{\text{co},\mathbf{p}}$ using the Euclidean distance in 2D image space as the distance metric.

6.5.4.2 *ARAP Loss*

Only using the above loss would lead to an ill-posed problem as vertices could drift along the visual hull carved by the silhouette images without receiving any penalty resulting in distorted meshes. Thus, an as-rigid-as-possible regularization (Sorkine and Alexa, 2007; Sumner et al., 2007) is employed, which is defined as

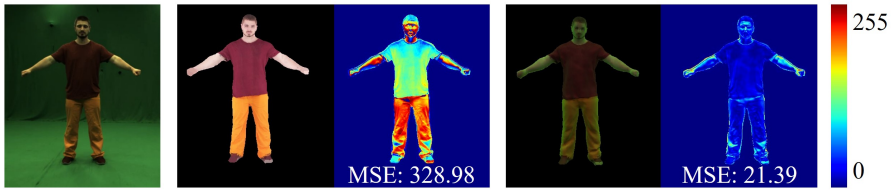
$$\mathcal{L}_{\text{arap}}(\mathbf{A},\mathbf{T}) = \sum_{k=1}^{K}\sum_{l\in\mathcal{N}_{\text{n}}(k)} u_{k,l}\|d_{k,l}(\mathbf{A},\mathbf{T})\|_1 \tag{6.9}$$

$$d_{k,l}(\mathbf{A},\mathbf{T}) = R(\mathbf{a}_k)(\mathbf{g}_l-\mathbf{g}_k)+\mathbf{t}_k+\mathbf{g}_k-(\mathbf{g}_l+\mathbf{t}_l).$$

Material-aware weighting factors $u_{k,l}$ (Habermann et al., 2020) are used, which are computed by averaging the rigidity weights s_i of all vertices attached to node k and l. Thus, different levels of rigidity are assigned to individual surface parts, e.g., graph nodes attached to skirt vertices can deform more freely than those attached to skin vertices.

6.6 LIGHTING ESTIMATION

So far, the posed and coarsely deformed character $\mathbf{V}_{\text{co}}$ can be obtained using EGNet and Equation 6.4. What is still missing are the finer



Ground Truth | Character **w/o** Explicit Light Modeling | Character **with** Explicit Light Modeling

Figure 6.4: From left to right. Ground truth image. Template rendered with the static texture. Mean squared pixel error (MSE) image computed from the masked ground truth and the rendering. Template rendered with the static texture and *the optimized lighting*. MSE image computed from the ground truth and the render with optimized lighting. Note that the optimized lighting clearly lowers the error between the rendering and the ground truth, which is advantageous for learning the per-vertex displacements. Further note that this is not the final appearance result. Instead, only the optimized lighting is used to improve the dense rendering loss, which supervises the displacement network that will be introduced in the next section.

deformations which are hard to capture just with multi-view silhouette images. Thus, it is aimed for a dense rendering loss that takes the posed and deformed geometry along with the static texture $\mathcal{T}_{\text{st}}$, renders it into all camera views, and compares it to the corresponding images. However, the lighting condition differs when capturing the scan of the subject, and therefore the texture and the lighting in the multi-camera studio can vary due to different light temperatures, camera optics and sensors, and scene reflections as shown in Figure 6.4. As a remedy, a differentiable rendering is proposed that also accounts for the difference in lighting and explicitly optimizes the lighting parameters for the multi-camera studio sequences.

6.6.1 *Differentiable Rendering*

It is assumed that the subject has a purely Lambertian surface reflectance (Lambert, 1760) and that the light sources are sufficiently far away, resulting in an overall smooth lighting environment. Hence, the efficient Spherical Harmonics (SH) lighting representation (Mueller, 1966) can be used, which models the scene lighting only with a few coefficients. To account for view-dependent effects, each of the C cameras has its own lighting coefficients $\mathbf{l}_c \in \mathbb{R}^{9\times 3}$ which in total sums up to $27C$ coefficients. It is assumed that the image has a resolution of $W \times H$. To compute the RGB color of a pixel $\mathbf{u} \in \mathcal{R}$ where $\mathcal{R} = \{(u, v)|u \in [1, W], v \in [1, H]\}$ in camera view c, the rendering function

$$\Phi_{c,\mathbf{u}}(\mathbf{V}, \mathcal{T}, \mathbf{l}_c) = a_{c,\mathbf{u}}(\mathbf{V}, \mathcal{T}) \cdot i_{c,\mathbf{u}}(\mathbf{V}, \mathbf{l}_c) \tag{6.10}$$

is used which takes the vertex positions $\mathbf{V}$, the texture $\mathcal{T}$, and the lighting coefficients $\mathbf{l}_c$ for camera c. As a Lambertian reflectance model is assumed, the rendering equation simplifies to a dot product of the albedo $a_{c,\mathbf{u}}(\mathbf{V}, \mathcal{T})$ of the projected surface and the illumination $i_{c,\mathbf{u}}(\mathbf{V}, \mathbf{l}_c)$. The albedo can be computed as

$$a_{c,\mathbf{u}}(\mathbf{V}, \mathcal{T}) = v_{c,\mathbf{u}}(\mathbf{V}) t_{c,\mathbf{u}}(\mathbf{V}, \mathcal{T}) \tag{6.11}$$

where $v_{c,\mathbf{u}}(\mathbf{V})$ is an indicator function that computes whether a surface is visible or not given the pixel position, camera, and surface. Like traditional rasterization (Pineda, 1988), $t_{c,\mathbf{u}}(\mathbf{V}, \mathcal{T})$ computes the barycentric coordinates of the point on the triangle that is covering pixel $\mathbf{u}$, which are then used to bi-linearly sample the position in texture map space. The lighting can be computed in SH space as

$$i_{c,\mathbf{u}}(\mathbf{V}, \mathbf{l}_c) = \sum_{j=1}^{9} \mathbf{l}_{c,j} SH_j(n_{c,\mathbf{u}}(\mathbf{V})) \tag{6.12}$$

where $\mathbf{l}_{c,j} \in \mathbb{R}^3$ are the jth SH coefficients for each color channel and SH_j are the corresponding SH basis functions. $n_{c,\mathbf{u}}(\mathbf{V})$ computes the screen space pixel normal given the underlying geometry.

Note that the final color $\Phi_{c,\mathbf{u}}(\mathbf{V}, \mathcal{T}, \mathbf{l}_c)$ only depends on the geometry $\mathbf{V}$, the texture $\mathcal{T}$, and the lighting coefficients $\mathbf{l}_c$ assuming camera and pixel position are fixed. As all the above equations (except visibility) are differentiable with respect to these variables, gradients can be backpropagated through the rendering process. The visibility $v_{c,\mathbf{u}}(\mathbf{V})$ is fixed during one gradient step.

6.6.2 *Lighting Optimization*

To optimize the lighting, it is assumed that the texture and geometry are fixed. Therefore, the texture is set to $\mathcal{T} = \mathcal{T}_{\text{st}}$ which is the static texture obtained from the scan. The geometry is set to $\mathbf{V} = \mathbf{V}_{\text{co}}$ which is the deformed and posed vertex positions regressed by EGNet. Now, the lighting coefficients $\mathbf{l}_{\text{mcs},c}$ for camera c can be computed by minimizing

$$\mathcal{L}_{\text{light}}(\mathbf{l}_{\text{mcs},c}) = \sum_{\mathbf{u} \in \mathcal{R}} \| \Phi_{c,\mathbf{u}}(\mathbf{V}_{\text{co}}, \mathcal{T}_{\text{st}}, \mathbf{l}_{\text{mcs},c}) - \mathcal{I}_{c,\mathbf{u}} \|^2 \tag{6.13}$$

for all frames of the training sequence. Note that all frames are used while the lighting coefficients are the same across frames. As one cannot solve for all frames jointly, stochastic gradient descent is employed, which randomly samples 4 frames and 30,000 iterations are applied. As a result, the optimal lighting coefficients $\mathbf{l}^*_{\text{mcs},c}$ are obtained, and the rendering with the static texture and the optimized lighting matches the global appearance much better than a rendering which is not explicitly modeling lighting (see Figure 6.4). As the lighting and the

Input Driving Motion | **Without** DeltaNet | **With** DeltaNet

Figure 6.5: From left to right. Input motion. The result without the displacements predicted by DeltaNet. The result with the predicted displacements. Note that the displacements clearly improve the overlay as they allow capturing finer geometric details.

texture are now known, they can be leveraged in the rendering function Equation 6.10 to densely supervise the per-vertex displacements, which are regressed on top of the embedded deformation parameters in the following.

6.7 VERTEX DISPLACEMENT REGRESSION

Next, the network architecture for regressing vertex displacement is introduced, and it is explained how the architecture is supervised.

6.7.1 *Displacement Network DeltaNet*

The goal is capturing also finer deformations, which the character representation models as per-vertex displacements $\mathbf{d}_i$, that were previously set to zero. The second network, called *DeltaNet*, takes the motion sequence again and regresses the displacements $\mathbf{D} \in \mathbb{R}^{N\times 3}$ for the N vertices of the template mesh in canonical pose. Here, the ith row of $\mathbf{D}$ contains the displacement $\mathbf{d}_i$ for vertex i. Similar to the EGNet, the pose is represented in the same space as the output space of the regression task. Thus, the template mesh is posed to the respective poses from the normalized motion $\hat{\mathcal{M}}$ using dual quaternion skinning resulting in $F+1$ consecutive 3D vertex positions, which is denoted as $\hat{\mathcal{M}}_{\text{delta}} \in \mathbb{R}^{N\times 3(F+1)}$. DeltaNet is denoted as the function $f_{\text{delta}}(\hat{\mathcal{M}}_{\text{delta}}, \mathbf{w}_{\text{delta}}) = \mathbf{D}$ where $\mathbf{w}_{\text{delta}}$ are the trainable network weights. Similarly, the displacement for a single vertex is referred to as $f_{\text{delta},i}(\hat{\mathcal{M}}_{\text{delta}}, \mathbf{w}_{\text{delta}}) = \mathbf{d}_i$ and the final posed and deformed character vertices are defined as

$$\mathcal{C}_i(\boldsymbol{\theta}, \boldsymbol{\alpha}, \mathbf{z}, f_{\text{eg}}(\hat{\mathcal{M}}_{\text{eg}}, \mathbf{w}_{\text{eg}}), f_{\text{delta},i}(\hat{\mathcal{M}}_{\text{delta}}, \mathbf{w}_{\text{delta}})) = \mathbf{v}_{\text{fi},i}. \tag{6.14}$$

$\mathbf{V}_{\text{fi}} \in \mathbb{R}^{N\times 3}$ denotes the matrix that contains all the posed and deformed vertices. Again, the SAGCN architecture is used as it is able to

preserve local structures better than fully connected architectures. The graph is defined by the connectivity of the template mesh, and each vertex is a graph node. The input is $\hat{\mathcal{M}}_{\text{delta}}$ and therefore the input and output feature sizes are $H_{\text{input}} = 3(F+1)$ and $H_{\text{output}} = 3$, respectively. Further, the hyperparameters are set to $H_1 = 16$, $L = 8$, and $R = 3$. Different from EgNet, the dense block is employed as the mesh graph is very large, and thus sharing features for very far nodes is difficult otherwise. Figure 6.5 shows that adding these displacements improves the silhouette matching as the finer geometric details can be captured.

6.7.2 *Weakly Supervised Losses*

The displacement predictions and therefore $\mathbf{V}_{\text{fi}}$ are weakly supervised using the loss function

$$\mathcal{L}_{\text{Delta}}(\mathbf{V}_{\text{fi}}) = \mathcal{L}_{\text{chroma}}(\mathbf{V}_{\text{fi}}) + \mathcal{L}_{\text{sil}}(\mathbf{V}_{\text{fi}}) + \mathcal{L}_{\text{lap}}(\mathbf{V}_{\text{fi}}) \tag{6.15}$$

which is composed of two multi-view image-based data terms and a spatial regularizer. $\mathcal{L}_{\text{sil}}$ is the silhouette loss introduced in Equation 5.11 but now applied to the vertices after adding the displacements to still ensure matching model and image silhouettes.

6.7.2.1 *Chroma Loss*

The silhouette-based loss alone can only constrain the boundary vertices of the model. But since one wants to learn the displacements per vertex, a denser supervision is required, and therefore a dense rendering loss

$$\mathcal{L}_{\text{chroma}}(\mathbf{V}_{\text{fi}}) = \sum_{c=1}^{C} \sum_{\mathbf{u} \in \mathcal{R}} \| g(\Phi_{c,\mathbf{u}}(\mathbf{V}_{\text{fi}}, \mathcal{T}_{\text{st}}, \mathbf{l}^*_{\text{mcs},c})) - g(\mathcal{I}_{c,\mathbf{u}}) \|^2 \tag{6.16}$$

is employed, which renders the mesh $\mathbf{V}_{\text{fi}}$ into the camera view c and compares it with the ground truth image $\mathcal{I}_c$ by using the differentiable renderer proposed in the previous section. In contrast to the previous rendering loss (see Equation 6.13), the color transform g is applied to both the rendered and the ground truth image. g converts RGB values into the YUV color space and only returns the UV channels. Thus, the loss is more invariant to shadow effects such as self-shadows which are not modeled by the renderer. Instead, the loss mainly compares the chroma values of the rendering and the ground truth.

6.7.2.2 *Laplacian Loss*

Only using the multi-view image-based constraints can still lead to distorted geometry. Thus, the posed and deformed model is further regularized with a Laplacian regularizer

$$\mathcal{L}_{\text{lap}}(\mathbf{V}_{\text{fi}}) = \sum_{i=1}^{N} s_i \| |\mathcal{N}_i| \left(\mathbf{v}_{\text{fi},i} - \mathbf{v}_{\text{co},i}\right) - \sum_{j \in \mathcal{N}_i} \left(\mathbf{v}_{\text{fi},j} - \mathbf{v}_{\text{co},j}\right) \|^2 \tag{6.17}$$

which ensures that the Laplacian of the mesh before and after adding the displacements is locally similar. Here, $\mathcal{N}_i$ is the set that contains the indices of the one ring neighbourhood of vertex i, and s_i are the per-vertex spatially varying regularization weights.

6.8 DYNAMIC TEXTURE REGRESSION

To add further realism to the poseable neural character, it is indispensable to have a realistic-looking texture. Although the scan provides a static texture, it is found that wrinkles are baked in and thus look unrealistic for certain poses, and further, it cannot account for view-dependent effects. Therefore, the goal is to also regress a motion and view point dependent texture $\mathcal{T}_{\text{dyn}} \in \mathbb{R}^{1024 \times 1024 \times 3}$.

As explained in Section 6.4, the normalized motion $\tilde{\mathcal{M}}$ is used as a conditioning input. Regressing textures just from these joint angles is difficult as the input and output are in different spaces. Thus, the mesh is posed according to the poses in $\tilde{\mathcal{M}}$ and the global normals are rendered into a texture map. By stacking the normal maps for each of the $F+1$ poses in $\tilde{\mathcal{M}}$, one obtains $\mathcal{T}_{\text{norm}} \in \mathbb{R}^{1024 \times 1024 \times 3(F+1)}$ where a texture size of 1024×1024 is used. As textural appearance does not only depend on poses but also on the positioning of the subject with respect to the camera, further the camera position and orientation is encoded into texture space denoted as $\mathcal{T}_{\text{cam}} \in \mathbb{R}^{1024 \times 1024 \times 6}$ where each pixel contains the position and orientation of the camera. By concatenating $\mathcal{T}_{\text{norm}}$ and $\mathcal{T}_{\text{cam}}$, $\mathcal{T}_{\text{input}} \in \mathbb{R}^{1024 \times 1024 \times 3(F+1)+6}$ is obtained, which is the final input to the texture regression network.

The texture network, *TexNet*, is based on the UNet architecture (Isola et al., 2017b), which is adapted to handle input and output dimensions of size 1024×1024. It takes the input texture encoding $\mathcal{T}_{\text{input}}$ and outputs the dynamic texture $\mathcal{T}_{\text{dyn}}$. Note that due to the input representation, the network can learn motion-dependent texture effects as well as view-dependent effects.

6.8.1 *Photometric Loss*

To supervise $\mathcal{T}_{\text{dyn}}$ for the conditioning camera view c', a texture loss

$$\mathcal{L}_{\text{texture}}(\mathcal{T}_{\text{dyn}}) = \sum_{\mathbf{u} \in \mathcal{R}} |\hat{\mathcal{F}}_{c',\mathbf{u}}(\Phi_{c',\mathbf{u}}(\mathbf{V}_{\text{fi}}, \mathcal{T}_{\text{dyn}}, \mathbf{I}_{\text{sh}}) - \mathcal{I}_{c',\mathbf{u}})| \tag{6.18}$$

is imposed, which renders the character using the dynamic texture regressed from TexNet and the geometry from the EgNet and DefNet and compares it to the real image. Here, $\hat{\mathcal{F}}_{c,\mathbf{u}}$ is the eroded image foreground mask. An erosion is applied to avoid that background pixels are projected into the dynamic texture if the predicted geometry does not perfectly align with the image. $\mathbf{I}_{\text{sh}}$ denotes the identity lighting. In contrast to the previous rendering losses, the network is only supervised on the conditioning view and not on all camera views.

6.9 EVALUATION

All the results are computed on a machine with an AMD EPYC 7502P processing unit and an Nvidia Quadro RTX 8000 graphics card. The proposed approach can run at 38 frames per second (fps) at inference time and therefore allows interactive applications, as discussed later. For the first frame of a test sequence, the pose of the first frame is copied over as the "previous frames" of the motion window as there are no real previous frames.

6.9.1 *Dataset*

A new dataset, called *DynaCap*, is created, which consists of 5 sequences containing 4 subjects wearing 5 different types of apparel, e.g., trousers and skirts (see Figure 6.6). Each sequence is recorded at 25fps and is split into a training and testing recording, which contain around 20,000 and 7,000 frames, respectively. The training and test motions are significantly different from each other. Following common practice, separate recordings are acquired for training and testing (instead of randomly sampling from a single sequence). For each sequence, the subject was asked to perform a wide range of motions like “dancing”, which was freely interpreted by the subject. 50 to 101 synchronized and calibrated cameras at a resolution of 1285×940 were used for the recording. Further, each person was scanned to acquire a 3D template, as described in Section 6.3, which is rigged to a skeleton. For all sequences, the skeletal motion is estimated using (*The Captury* 2020), and the foreground is segmented using color keying. The new dataset is publicly available, as there are no other datasets available that target exactly such a setting, namely a single actor captured for a large range of motions and with such a dense camera setup. The dataset can be particularly interesting for dynamic neural scene representation approaches and can serve as a benchmark.

In addition, the subjects *S1*, *S2*, and *S4* of the publicly available DeepCap dataset (Habermann et al., 2020) are used who wear trousers, T-shirts, skirts, and sleeves to evaluate the proposed method also on external data, which has a sparser camera setup. The dataset comes

Figure 6.6: DynaCap dataset. 5 subjects wearing different types of apparel are recorded with multiple calibrated and synchronized cameras. Further, a 3D template mesh is captured and rigged to a skeleton. For each frame, the ground truth 3D skeletal pose, as well as ground truth foreground segmentation, are computed.

along with ground truth pose tracking, calibrated, segmented, and synchronized multi-view imagery, in addition to a rigged template mesh. The dataset contains between 11 and 14 camera views at a resolution of 1024×1024 and a frame rate of 50fps.

6.9.2 *Qualitative Results*

In Figure 6.7, results for all 8 sequences are illustrated, showing different types of apparel. Again, note that the presented method learns video realistic motion- and view-dependent dynamic surface deformation, including also deformations of loose apparel (such as the skirt and dress in Figure 6.7), without requiring a physics simulation, and texture *only* from multi-view imagery and does *not* require any dense 3D data such as depth maps, point clouds or registered meshes for supervision. The proposed approach works not only well for tighter clothes such as pants but also for more dynamic ones like skirts. It is

demonstrated that the proposed approach can create video realistic results for *unseen* very challenging and fast motions, e.g., jumping jacks. Moreover, the texture is consistent while changing the viewpoint (images without green screen background), which shows that the view conditioned TexNet also generalizes to novel viewpoints. The generalization comes from the fact that the networks for deformation regression as well as for texture regression focus on local configurations rather than the full 3D body motion. However, the network still allows global reasoning, but this effect is dampened by the network design. Technically, this is accomplished by the local graph/image features and the graph/image convolutions. Further, note the view-dependent effects like reflections on the skin and clothing (second and third column where the pose is kept fixed and only the viewpoint is changed). Given an image of the empty capture scene (images with green screen background), the presented approach allows augmenting the empty background image with the obtained results to produce realistic-looking images.

Figure 6.8 shows that the predicted geometry (on test data) precisely overlays to the corresponding image, which demonstrates that the presented approach generalizes well to unseen motions. Importantly, the ground truth frame showing the actor is *not* an input to the proposed method as it only takes the skeletal motion, which is extracted from the video. The textured result is also shown as an overlay onto the ground truth. The TexNet generalizes well to unseen motions, captures the motion-dependent dynamics, and looks photo-realistic as ground truth and the rendered result look almost identical.

6.9.3 *Comparison*

Only a few people in the research community have targeted creating video realistic characters from multi-view video that can be controlled to perform unseen skeleton motions. There are only three previous works (Casas et al., 2014; Shysheya et al., 2019; Xu et al., 2011) that also assume multi-view video data for building a controllable and textured character. However, these works do not provide their code and thus are hard to compare to. Moreover, they either do not share their data (Shysheya et al., 2019) or the publicly available sequences are too short for training the proposed approach (Casas et al., 2014; Xu et al., 2011) and as well lack a textured template, which the presented method assumes as given. Therefore in Table 6.1, it is resorted to a conceptual comparison showing the advantage of the proposed method.

The earlier works of (Xu et al., 2011) and (Casas et al., 2014) are both non learning-based and instead use texture retrieval to synthesize dynamic textures. In contrast to the presented approach, they both suffer from the fact that their geometry is either fully driven by skinning-

Figure 6.7: Qualitative results. From left to right. Input testing pose and the posed, deformed, and textured result shown from an arbitrary viewpoint. The obtained result for another testing pose and viewpoint. The same pose as in the second column but rendered from a different viewpoint. Note the view-dependent appearance change on the skin and clothing due to view-dependent reflections. The last two columns show testing poses but viewed from the training camera viewpoints. This allows augmenting the empty background that is captured for each camera with the result of the proposed method.

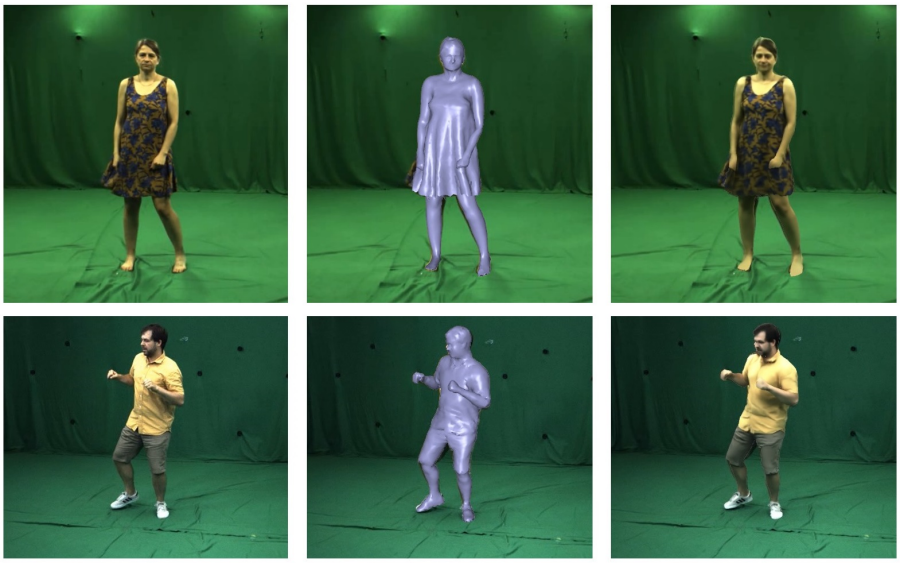


Figure 6.8: The geometry and texture networks generalize well to unseen motions as the geometry overlays nicely onto the ground truth frame, and the final textured result looks almost identical to the ground truth. Importantly, the presented method does *not* take the ground truth frame as an input as it only takes the unseen motion from the video.

based deformations (Xu et al., 2011) or by motion graphs (Casas et al., 2014). Thus, they cannot model motion-dependent geometric deformations and fail to model plausible dynamics of loose apparel, as the proposed method can do it. Moreover, as they rely on retrieval-based techniques, their approaches do not generalize well to motions different from motions in the dataset. Furthermore, the retrieval is expensive to compute, making real-time applications impossible. In contrast, the proposed approach leverages dedicated geometry networks (EGNet and DeltaNet), which predict motion-dependent geometry deformations for both tight and loose apparel. Further, the presented method enables animation and control in real time and generalizes well to unseen motions (see Figure 6.7).

More recently, Textured Neural Avatars (Shysheya et al., 2019) was proposed as the first learning-based approach for creating controllable and textured characters using multi-view data. In contrast to the proposed approach, they do not model geometry explicitly but use DensePose (Güler et al., 2018) as a geometric proxy in image space. As a consequence, their approach does not provide space-time coherent geometry as well as motion-dependent surface deformation, which is important in most graphics and animation settings. Moreover, they recover a static texture during training, which prevents modelling motion- and view-dependent effects.

Comparison to Previous Multi-view Based Methods				
Property	Xu et al.	Casas et al.	Shysheya et al.	**Ours**
Dyn. Geo.	✗	✗	✗	✓
Dyn. Tex.	✓	✓	✗	✓
View Dep.	✓	✓	✗	✓
Control	✓	✓	✓	✓
Real-time	✗	✗	✓	✓
Unseen Motions	✗	✗	✓	✓
Loose Clothing	✗	✗	✗	✓

Table 6.1: Conceptual comparison to previous multi-view based approaches for controllable character animation / synthesis (Casas et al., 2014; Shysheya et al., 2019; Xu et al., 2011). Note that all previous works fall short in multiple desirable categories while the proposed approach fulfills all these requirements.

6.9.4 *Quantitative Evaluation*

Next, the proposed approach is quantitatively evaluated and compared to previous state-of-the-art methods in terms of geometric and appearance accuracy.

6.9.4.1 *Geometry*

To evaluate the proposed approach in terms of geometry, the challenging *S4* testing sequence (11,000 frames) of the DeepCap dataset (Habermann et al., 2020) is leveraged and shown in the top row of Figure 6.8. The model is trained on the corresponding multi-view training sequence, and their mesh template is used. The evaluation procedure follows the one described in the original paper. Therefore, the multi-view foreground mask overlap between ground truth foreground segmentation and the foreground mask obtained from the projected and deformed model is measured on all available views (AMVIoU) and averaged over every 100th frame.

In Table 6.2, the proposed approach is compared to the multi-view baseline implementation of (Habermann et al., 2020), referred to as MVBL. Here, they perform optimization-based multi-view pose and surface fitting using sparse and dense image cues, e.g., 2D joint predictions and the foreground masks. Importantly, they apply this on the *testing* sequence directly, whereas the proposed method only takes the skeletal motion without even seeing the multi-view imagery. Nonetheless, the obtained results are more accurate than MVBL. It is found that their sequential optimization of pose and deformation can fall into erroneous local optima, resulting in a worse overlay. In contrast, the

AMVIoU (in %) on S4 sequence	
Method	**AMVIoU↑**
MVBL (Habermann et al., 2020)	88.14
(Kavan et al., 2007)	79.45
Ours	**90.70**
Ours (Train)	94.07

Table 6.2: Accuracy of the surface deformation. Note that the proposed method outperforms the pure skinning-based approach (Kavan et al., 2007) as they cannot account for dynamic cloth deformations. The presented method further improves over MVBL even though this optimization-based approach sees the multi-view test images. Finally, the proposed approach performs similarly on training and testing data showing that the geometry networks generalize to unseen motions.

presented method benefits from the randomness of the stochastic gradient descent and the shuffling of data, which reduces the likelihood of getting stuck in local optima. The poseable and dynamic representation is also compared to the classical Dual Quaternion character skinning (Kavan et al., 2007) where the same poses, which are also used by the proposed approach, are leveraged to animate the rigged character. Skinning can merely approximate skeleton-induced surface deformation, but it fails to represent dynamic clothing deformations, as the proposed method can handle them. Thus, the presented approach clearly outperforms their approach as they cannot account for the surface deformation caused by the motion of the actor, e.g., swinging of a skirt.

The same metrics are also reported on the training data. Even from a reconstruction perspective, the proposed method produces accurate results during training, and the proposed representation is able to fit the image observations almost perfectly. Notably, there is only a small accuracy difference between training and testing performance. This confirms that the presented approach generalizes well to unseen motions.

6.9.4.2 *Texture*

In Table 6.3, the realism of the motion-dependent dynamic texture is evaluated on the same sequence as before (testing sequence of *S4*). It is again trained on the training motion sequence of *S4*, but camera 4 is hold out as a test view. The presented approach is evaluated on *Train Camera* 0 and *Test Camera* 4 for *Train Motions* and *Test Motions*. Therefore, the mean squared image error (MSE) and the structural similarity index measure (SSIM) between the rendered model and the ground truth multi-view images are computed and averaged over every

Photometric Error on *S4*		
Method	**MSE ↓**	**SSIM↑**
Ours (Train Motion / Train Camera)	14.79	0.99054
Ours (Train Motion / Test Camera)	31.44	0.98610
Ours (Test Motion / Train Camera)	29.00	0.98357
Ours (Test Motion / Test Camera)	43.29	0.98278

Table 6.3: Photometric error in terms of MSE and SSIM averaged over every 100th frame. Note that the presented approach achieves overall low MSE results and high SSIM values. While the accuracy differs between test and train, the absolute accuracy is still comparably high, and the visual quality only decreases slightly, proving the generalization ability of the presented approach.

100th frame where the background is masked out as the proposed approach does not synthesize the background. The proposed method produces visually plausible results for novel motions rendered from a training view (see top row of Figure 6.8 and the 4*th* and 5*th* column of the second last row of Figure 6.7). But also for novel motions *and* novel camera views, the presented approach produces video-realistic results (see 1*th*, 2*th*, and 3*th* column of the second last row of Figure 6.7). Table 6.3 also quantitatively confirms this since all configurations of training/testing poses and camera views have a low MSE value and a high SSIM value. While there is an accuracy drop between test and train, visually, the quality only decreases slightly, and the absolute accuracy for each configuration is comparably high.

6.9.5 *Ablation*

Next, an ablation study is performed on the deformation and texture modules as well as the amount of required data.

6.9.5.1 *Deformation Modules*

First, the design choices for predicting the surface deformations are evaluated. Therefore, the impact of the DeltaNet is compared against only using EGNet, which is referred to as *EGNet-only*. Table 6.4 clearly shows that the additional vertex displacements improve the reconstruction accuracy as they are able to capture finer wrinkles and deformations (see also Figure 6.5). While the regressed embedded deformation still performs better than a pure skinning approach, it cannot completely match the ground truth silhouettes due to the limited graph resolution causing the slightly lower accuracy compared to using the displacements.

Further, the impact of the SAGC is evaluated and compared to two baselines using a fully connected (FC) architecture and an unstructured graph convolutional operator (Defferrard et al., 2016) where the latter is integrated into the overall architecture and therefore just replaces the SAGC operators. EGNet and DeltaNet are replaced with two fully connected networks that take the normalized motion angles as input, apply 19 fully connected layers (same depth as the proposed architecture) with nonlinear Elu activation functions, and output graph parameters and vertex displacements, respectively. As the fully connected networks have no notion of locality, they are not able to generalize well. Further, one can see that the proposed graph convolutional operation performs better than the one proposed by (Defferrard et al., 2016) because the latter shares weights across nodes while the presented approach uses node-specific weights, which are able to encode the underlying knowledge about the individual deformation behaviours of the surface.

The importance of predicting the per-vertex displacements in the canonical pose space is also evaluated and compared to predicting them in the global pose space. Note that the disentanglement of pose and deformation helps with the generalization of the network, which leads to better accuracy in terms of foreground overlay.

Finally, the impact of the chroma loss is evaluated and compared to only using silhouette supervision. Note that reporting the IoU would not be meaningful as the silhouette loss alone can already ensure matching silhouettes. However, drifts along the visual hull, carved by the silhouette images, cannot be well tracked by the silhouette term alone, as shown in Figure 6.9. The chroma loss penalizes these drifts, both during training and testing, leading to better results. This can be best evaluated by comparing the MSE of the deformed model with the static texture and the ground truth image as shown in Figure 6.9 Here, using the chroma loss has an error of 38.20 compared to an error of 44.53 when only the silhouette loss is used during test time. This clearly shows that the chroma error can disambiguate drifts on the visual hull and thus gives more accurate results.

6.9.5.2 *Texture Module*

Next, the motion- and view-dependent texture is compared to using a static texture rendered with and without optimized lighting. Using the static texture without optimized lighting leads to the highest error. Optimizing the light already brings the rendering globally a bit closer to the ground truth image but still fails to represent important dynamic and view-dependent effects. By applying the dynamic texture also motion- and view-dependent texture effects can be captured, resulting in the lowest error.

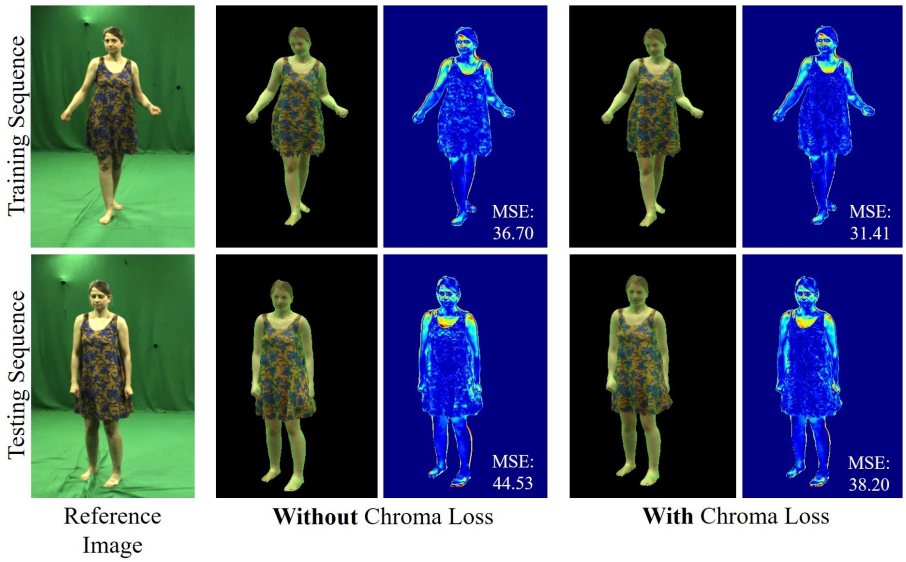


Figure 6.9: Impact of the chroma loss. During training and testing, the chroma loss disambiguates drifts on the visual hull and gives more accurate results.

6.9.5.3 *Amount of Data*

Finally, the influence of the number of training cameras is evaluated for the *OlekDesert* sequence in Table 6.5. More precisely, the training with 5, 10, 25, and 49 cameras placed around the scene in a dome-like arrangement is tested here. The respective test motions are used for all reported metrics. For computing the MSE, camera 46 is chosen, which was not part of the training views for all experiments. Note that already 5 cameras can lead to plausible results. Interestingly, with such a sparse setup, the presented approach still produces coherent results for unseen viewpoints as the prediction is in canonical texture space, which implicitly regularizes the predictions, leading to a better generalization ability. However, adding more cameras further improves both geometry and texture quality.

6.9.6 *Applications*

As shown in Figure 6.10, the presented method can be used in several applications such as motion re-targeting where a source actor (blue dress girl) drives the character model (red shirt girl). Further, the proposed method synthesizes new free-viewpoint videos of an actor only with a driving motion sequence. Moreover, an interactive interface is implemented, where the user can freely change the skeletal pose and 3D camera viewpoint, and the method produces the posed, deformed, and texture geometry in real time.

Ablation on the S4 sequence		
Method	**AMVIoU↑**	**MSE↓**
EGNet-only	87.89	—
Fully Connected	87.48	—
Unstructured GraphConv	83.30	—
Global Pose Space	89.81	—
Without Lighting and Dynamic Texture	—	176.99
Without Dynamic Texture	—	60.50
Ours	**90.70**	**43.29**

Table 6.4: Ablation study. The design choices are evaluated for the geometry networks and texture networks. Note that the proposed approach beats the baselines in all aspects, confirming that the design choices indeed lead to an improvement.

Ablation on the OlekDesert sequence		
Method	**AMVIoU↑**	**MSE↓**
5 camera views	90.27	20.85
10 camera views	90.34	19.32
25 camera views	90.36	17.49
Ours (49 views)	**90.68**	**16.72**

Table 6.5: Influence of the number of available training cameras. Already with few cameras, the proposed method achieves plausible results. However, adding more cameras further improves the quality of both geometry and texture.

6.10 LIMITATIONS AND FUTURE WORK

The presented approach approximates clothing dynamics in a data-driven and plausible way, but actual physics-based clothing animation may still lead to further improved results. In future research, this could be handled by employing those physics-based priors in the learning process or even at inference. Further, the proposed method cannot handle apparent topological changes such as taking off pieces of apparel as well as smaller-scale wrinkles. The current progress in implicit representations combined with the proposed representation could help to generate such changes and details even though they are radically different from the initial template mesh. The facial expression and hands are not tracked. 2D face landmark trackers, as well as hand trackers, could be used to also track hands and face so that they can also be controlled in the deep dynamic character. Currently,

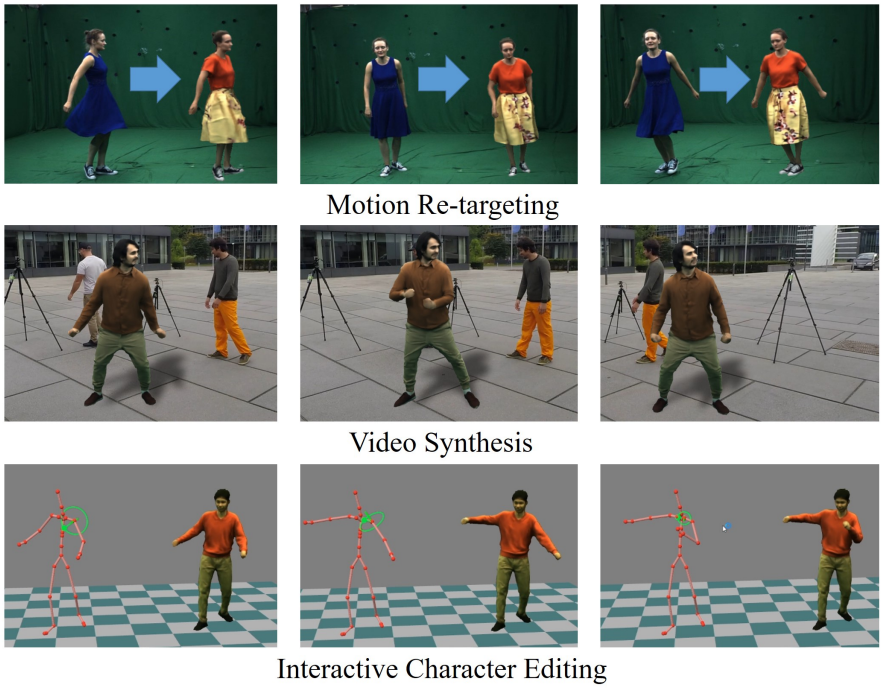


Figure 6.10: Applications. The proposed method can be used in several applications such as motion re-targeting, neural video synthesis, and interactive character editing. Note that for all these applications, the proposed method produces video realistic results creating an immersive experience.

the training time of the network modules is quite long. In the future, more efficient training schemes could be explored to solve this issue. Moreover, the method relies on good foreground segmentation results. In consequence, the proposed method might receive a wrong silhouette supervision when multiple people or other moving objects, which are detected as foreground, are in the training scene. Explicitly modeling multi-person scenes and using a learning-based multi-person detector could help here. Finally, severely articulated poses like a handstand, which are not within the training motion distribution, can lead to wrong deformation and texture predictions.

6.11 CONCLUSION

This chapter presented a real-time method that enables animation of the dynamic 3D surface deformation and texture of highly realistic 3D avatars in a user-controllable way. Skeleton motion can be freely controlled, and avatars can be free-viewpoint rendered from any 3D viewpoint. To this end, a learning-based architecture was proposed, which regresses not only dynamic surface deformations but also dynamic textures. The proposed approach does not require any ground truth 3D supervision. Instead, it only needs multi-view imagery, and

it employs new analysis-by-synthesis losses for supervision. The obtained results outperform the state of the art in terms of surface detail and textural appearance, and therefore the high visual quality of the animations opens up new possibilities in video-realistic character animation, controllable free-viewpoint video, and neural video synthesis.

As discussed in the limitations of this chapter, the proposed method can only partially capture finer geometric details, and it is not able to handle topological changes. To this end, the next section presents ongoing work on character synthesis where the key idea is to combine implicit and explicit mesh representations to further enhance the geometric details of the deforming mesh and to improve the synthesis quality.

6.12 TOWARDS HIGHER FIDELITY 3D CHARACTER SYNTHESIS

As mentioned before, it remains challenging for the previously presented approach, also referred to as *DDC* in the following, to nicely capture the finer geometric details such as wrinkles on the clothing, especially if these wrinkles are very different from the ones that are baked in the initial (unposed and undeformed) template mesh (see Figure 6.11). To this end, this section introduces an ongoing project, which aims at further improving the motion-dependent geometric details of the motion- and view-dependent controllable 3D character as well as further improving the video synthesis quality. The key idea to achieve this goal is to combine an explicit mesh representation (based on Habermann et al., 2021a) with an implicit neural radiance field (NeRF) (Mildenhall et al., 2020) that surrounds the mesh. More precisely, the method takes a skeletal motion as input and predicts a motion-dependent deforming geometry as well as a motion- and view-dependent neural radiance field that is parameterized near the mesh. This has the advantage that the deformed and posed geometry can act as an initializer for the sampling and the feature accumulation of the neural radiance field, which allows more efficient sampling of the NeRF and, most importantly, enables NeRF to work also for dynamic scenes. However, not only the implicit component can be improved based on the explicit mesh, but the implicit representation can also be leveraged to further improve the regressed explicit mesh representation. It can be observed that the accumulated depth of the neural radiance field contains finer geometric details such as cloth wrinkles and can thus be used to guide the deformations of the explicit geometry, which enables the approach to be trained only with 2D multi-view imagery, without any ground truth 3D supervision. The approach is designed to be fully differentiable, allowing end-to-end training. Preliminary results show that the proposed method further improves the geometric quality while the high synthesis quality of previous work is preserved.

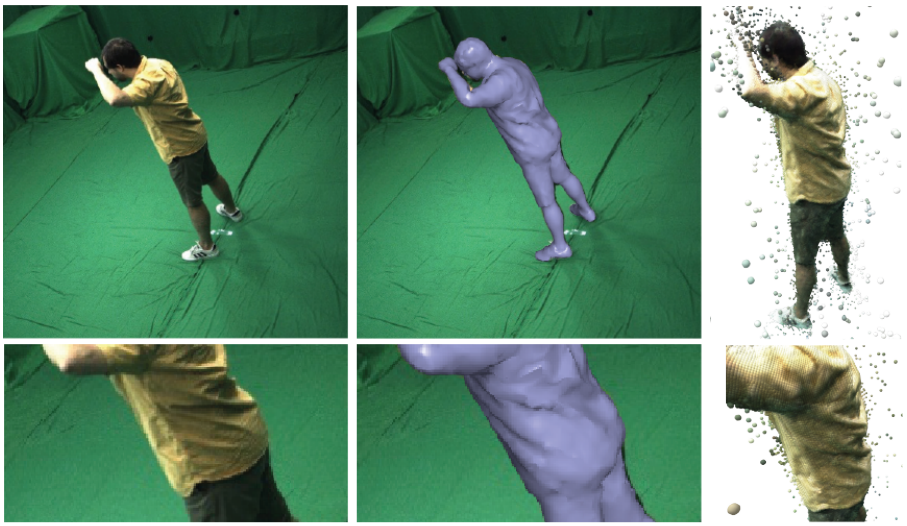


Figure 6.11: Comparison of the wrinkle accuracy of the previously presented method (Habermann et al., 2021a) and the mesh-conditioned NeRF, which is presented in this section, on the training sequence of the *Vlad* sequence of the DynaCap dataset (Habermann et al., 2021a). NeRF is trained on the single frame. Note that the mesh does not capture the geometric details while the NeRF volume contains the smaller scale wrinkles in the accumulated point cloud.

6.12.1 *Overview*

The input to the method is a skeletal motion and a camera pose, and it outputs a posed and deformed mesh as well as an attached neural radiance field which synthesizes the appearance of the subject. Figure 6.12 shows an overview of the proposed method. In the following, first some fundamentals regarding neural radiance fields (Mildenhall et al., 2020) are provided (Section 6.12.2). Then, the combined explicit and implicit mesh representation is introduced (Section 6.12.3), followed by a more detailed explanation of the sampling strategy (Section 6.12.3.1). Next, the network architecture is described (Section 6.12.3.2) and how it is supervised (Section 6.12.3.3). Then, it is explained how the entire approach is trained (Section 6.12.4), and some preliminary results are presented (Section 6.12.5). Finally, remaining challenges are discussed (Section 6.12.6).

6.12.2 *Background*

A neural radiance field (Mildenhall et al., 2020) is a deep, volumetric scene representation of a static scene, which allows for photo-realistic novel view synthesis. In detail, when one wants to render an image,

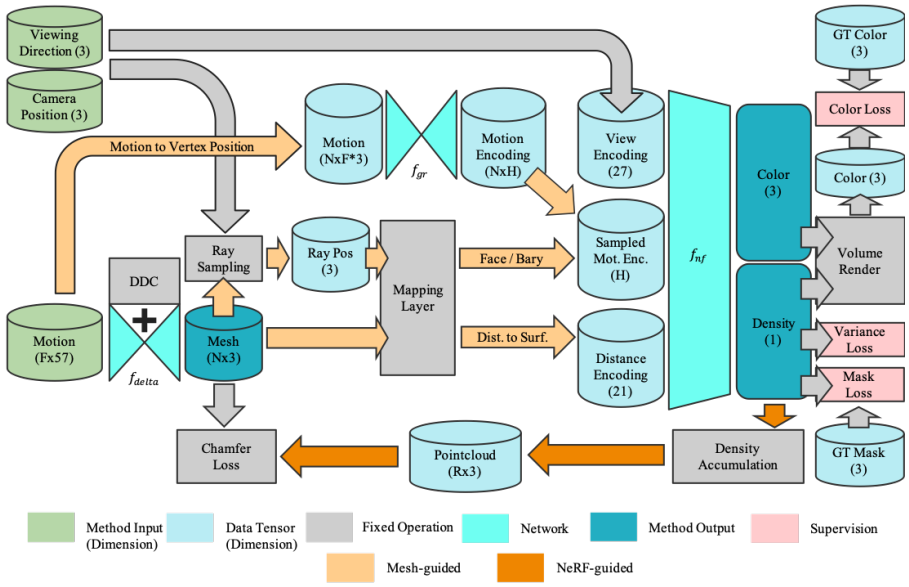


Figure 6.12: Overview of the proposed approach. The numbers in brackets denote the respective tensor sizes. It takes the camera parameters and the current motion of the kinematic skeleton as input. Then, a graph convolutional network predicts a motion-dependent posed and deformed mesh. Further, a NeRF-like architecture (Mildenhall et al., 2020) predicts a volume density and color per ray, which can be rendered to the final image of the subject using volume rendering. Importantly, the NeRF synthesis can be guided by the mesh (light orange arrows), which allows for synthesis of dynamic scenes and a better sampling strategy. As the accumulated depth of NeRF can be used to compute an accurate point cloud of the scene, this point cloud can also be used to refine the deformed and posed mesh (dark orange arrows). This process can be iteratively performed so that the implicit and explicit mesh representations improve each other.

each pixel is represented by a ray that has a normalized direction $\mathbf{d} \in \mathbb{R}^3$ and an origin $\mathbf{o} \in \mathbb{R}^3$. Then, $i \in \{0, ..., K\}$ samples along the ray at position $\mathbf{x}_i = \mathbf{o} + t_i \mathbf{d}$, where $\mathbf{x}_i \in \mathbb{R}^3$ and t_i is the depth along the ray, are drawn and fed into an MLP

$$f_{\text{nerf}}(\gamma(\mathbf{x}_i), \gamma(\mathbf{d}), \mathbf{w}_{\text{nerf}}) = (\mathbf{c}_i, \sigma_i) \tag{6.19}$$

which takes the positional encoding (Mildenhall et al., 2020) $\gamma(\cdot)$ of $\mathbf{x}_i$ and $\mathbf{d}$ as input. Then, the network predicts a color $\mathbf{c}_i \in \mathbb{R}^3$ and a density value $\sigma_i \in \mathbb{R}$. Here, $\mathbf{w}_{\text{nerf}}$ are the trainable network weights.

To obtain the final pixel color, the individual colors and densities are accumulated using volume rendering (Levoy, 1990) according to

$$\tilde{\mathbf{c}} = \sum_{i=0}^{K} T_i \alpha_i \mathbf{c}_i \tag{6.20}$$

$$T_i = \prod_{j=0}^{i-1} (1 - \alpha_j) \tag{6.21}$$

$$\alpha_i = 1 - e^{-\delta_i \sigma_i} \tag{6.22}$$

where δ_i is the Euclidean distance between the sample points $\mathbf{x}_{i+1}$ and $\mathbf{x}_i$.

Conveniently, NeRF can be supervised with multi-view images alone. Assuming C images of calibrated cameras are given as input, a pixel r from the images is chosen where the ground truth color $\mathbf{c}_{\text{gt}}^{(r)}$ is known. NeRF employs an $L2$ loss on the color difference

$$\mathcal{L}_{color}(\tilde{\mathbf{c}}^{(r)}) = \|\tilde{\mathbf{c}}^{(r)} - \mathbf{c}_{\text{gt}}^{(r)}\|^2 \tag{6.23}$$

where $\tilde{\mathbf{c}}^{(r)}$ is obtained according to Equation 6.20.

NeRF has shown state-of-the-art synthesis quality on static scenes and outperformed other scene representations. Interestingly, when training NeRF on a scene containing a human, the recovered depth maps show detailed wrinkle patterns despite some noise and outliers (see Figure 6.11). However, it is also important to note that the MLP has to be evaluated for each pixel of the image individually, which makes this computation slow, and the original NeRF is limited to static scenes and cannot handle dynamic ones. Further, since in the original approach, no prior knowledge about the scene is assumed, the sampling process has to cover the entire 3D space where potentially many sample points are in the empty space and thus do not contribute to the final color. In the following, a combination of a mesh-based representation and a neural radiance field is introduced, which leverages the strength of NeRF, namely the synthesis quality and the recovered depth maps, while most importantly allowing for user-controlled dynamic scenes of articulated humans that can be rendered from arbitrary views and that are efficient to compute and sample.

6.12.3 *Combined Explicit and Implicit Character Representation*

The main insight of the proposed geometry-based character representation of DDC previously presented in Equation 6.14 is that it can represent dynamically moving humans and that the reconstructed and simulated geometry is close to the ground truth. However, there still remains a residual in terms of surface accuracy. As a result, smaller wrinkles are not captured by the deformed geometry. One can think of this residual as an uncertainty volume around the deformed geometry. Thus, both representations, NeRF and DDC, have complementary advantages and disadvantages, and the aim in the following is to combine the two representations.

6.12.3.1 *Geometry-guided Sampling*

To this end, it is assumed that a pre-trained version of DDC is given as described in Section 6.7, which provides the posed and densely

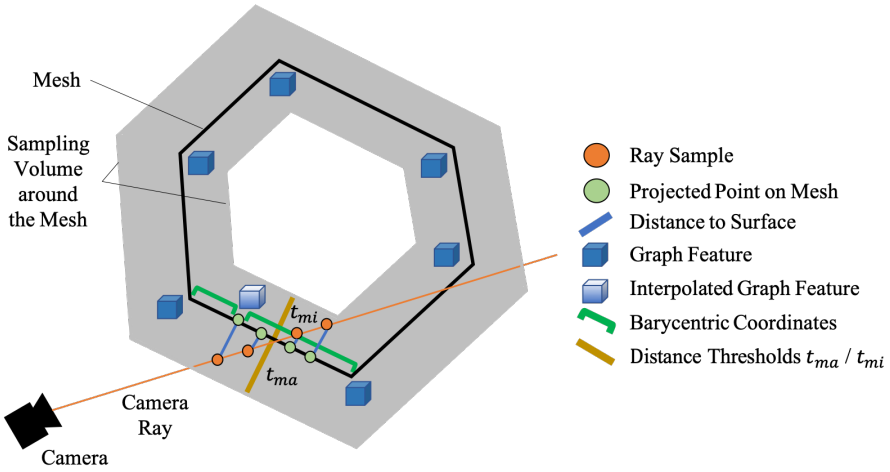


Figure 6.13: Illustration of the geometry-guided neural radiance field sampling. Note that samples only have to be drawn close to the mesh surface inside the volume marked in grey. Then, the distance and the barycentric coordinates are computed for interpolating the graph features that are attached to the mesh vertices.

deformed vertices $\mathbf{V}_{\text{fi}}$ of the template mesh. For simplicity, the index is dropped, and the template is defined as $\mathbf{V} = \mathbf{V}_{\text{fi}}$ in the following. Further, it is assumed that multi-view images of calibrated cameras and the respective foreground masks, which contain the actor, are given. Since reconstructing the background is not a goal of this approach, the images are masked using these foreground masks. Now, to combine the geometry with a NeRF, it has to be noted that the ith sample $\mathbf{x}_i$ along the ray of the NeRF can be represented as a function of the geometry

$$\begin{aligned} t_i^{(r)}(\mathbf{V}) = & \, er(\Phi_{\text{dp}}^{(r)}(\mathbf{V})) - t_{\text{mi}} \\ & + \frac{i}{K}\left((di(\Phi_{\text{dp}}^{(r)}(\mathbf{V})) + t_{\text{ma}}) - (er(\Phi_{\text{dp}}^{(r)}(\mathbf{V})) - t_{\text{mi}})\right) \end{aligned} \tag{6.24}$$

by replacing the depth t_i along the ray with $t_i^{(r)}(\mathbf{V})$ in the sampling formulation. Here, $\Phi_{\text{dp}}^{(r)}(\mathbf{V})$ is similar to the previously introduced differentiable rasterizer Φ with the difference that it renders the depth of the mesh with respect to the camera, and r indicates the specific pixel that was rendered in any of the available camera views. The function $di(\cdot)$ represents the dilation operator, which computes the maximum depth value in the depth map around the sampled location r. Similarly, $er(\cdot)$ computes the eroded value or minimum value around the sampled location r. The erosion and dilation ensure that the neural scene representation is also sampled on the foreground when the underlying mesh is erroneously not overlaying the ground truth foreground mask. Moreover, t_{mi} defines the volume that is sampled in front of the actual

surface, and similarly t_{ma} defines the volume that is sampled behind the actual surface by ensuring that the distance between the mesh and the sample point does not exceed t_{mi} and t_{ma}. Lastly, K defines the number of samples along the ray. When sampling r, only pixels that project onto the eroded/dilated depth maps are considered. Otherwise, they are discarded during the NeRF evaluation described later.

This allows a more effective sampling of the neural radiance field since most samples are very close to the actual surface. A visualization of this process is shown in Figure 6.13. Importantly, all terms (except the visibility in the depth renderer) of Equation 6.24 are differentiable with respect to the surface vertices, and thus the loss functions, which are employed on the 3D sample points, can also backpropagate into the deformation networks.

6.12.3.2 *Geometry-guided Motion Feature Assignment*

The other important property, which is missing in the original NeRF approach, is that it can only render a *static* scene under novel views. However, the proposed approach targets synthesizing novel views and performances of *dynamic* scenes. Fortunately, the posed and deformed DDC template can also help to enable the synthesis of dynamic scenes as well as improving the inference speed. Following the notations from earlier sections, the index of the frame is dropped, and it is assumed to be fixed for now.

The main idea is that motion-dependent deep features can be attached to the mesh, which are then the input to the NeRF instead of the positional encodings of the positions in global space (see Figure 6.13). More specifically, Equation 6.19 is modified as

$$f_{\text{nf}}(b(\mathbf{V}, \mathbf{x}_i, f_{\text{gr}}(\tilde{\mathcal{M}}_{\text{delta}}, \mathbf{w}_{\text{gr}})), \gamma(d(\mathbf{V}, \mathbf{x}_i)), \gamma(\mathbf{d}), \mathbf{w}_{\text{nf}}) = (\mathbf{c}_i, \sigma_i). \quad (6.25)$$

Here, $f_{\text{gr}}(\tilde{\mathcal{M}}, \mathbf{w}_{\text{gr}})$ is a graph convolutional network based on the architecture introduced in Section 6.5 which takes the translation normalized motion window $\tilde{\mathcal{M}}_{\text{delta}}$ encoded as posed vertex positions for the current frame as input and predicts a motion-dependent feature vector of size H per node of the graph. The template itself serves as the graph in this case. Then, $b(\cdot)$ takes the mesh $\mathbf{V}$, the sample $\mathbf{x}_i$ along the ray, and the per-vertex graph features as input and computes the closest point from $\mathbf{x}_i$ onto the mesh, which can be described by the three vertices enclosing the face and the respective barycentric coordinates. The latter ones are then used to interpolate the per-vertex graph features of the enclosing vertices resulting in a single feature vector also of size H. $d(\cdot)$ takes again the mesh and the sample point as input and outputs the signed and normalized distance between the two. Here, the interior of the mesh is defined to have a negative sign, and the points outside the mesh have a positive sign. The term *normalized* means that the actual distance is divided either by t_{mi} or

by t_{ma}, depending on whether the sample point is inside or outside the mesh surface. A positional encoding (Mildenhall et al., 2020) is then applied to the distance value. Finally, the modified NeRF MLP also takes the positional encoding of the viewing direction similar to the original approach. Importantly, the template mesh enables a more efficient sampling. Therefore, importance sampling and a second finer network branch are not required compared to the original NeRF architecture (Mildenhall et al., 2020).

This reformulation allows the network to encode the dynamic motion of the actor and thus allows NeRF to handle dynamically moving humans. Due to the specific motion representation and the graph convolutional architecture, the motion can be encoded locally on the surface level to improve the generalization. Moreover, the feature encoding at graph level only requires a single inference pass independent of the number of pixels. Thus, most of the capacity can be shifted into the graph convolutional architecture, which is faster to evaluate, and a shallower MLP can be used for the NeRF evaluation. This allows for a faster inference speed while still being able to synthesize high-quality results.

6.12.3.3 *NeRF-guided Geometry Supervision*

So far, it was only discussed how the NeRF representation can leverage the advantages of the underlying 3D template mesh; however, the geometry can also be improved using the NeRF. The key observation is that a weakly supervised setup, as presented in Section 6.7, struggles with recovering the finer wrinkles on the clothing. This is mostly due to the limited supervision from the rendering loss. Specifically, there are three reasons: 1) the rendering loss is very sensitive to local minima as gradients of the input image are computed with finite differences on the ground truth image; 2) this loss struggles with deformations that are out of the camera plane, and 3) the rendering loss cannot account for shadow and view-dependent effects. A possible solution is to further improve the rendering loss by using ray-tracing based approaches that are differentiable (Li et al., 2018; Nimier-David et al., 2019). However, their runtime is rather slow, which prevents them from being used in this setup. Fortunately, it can be observed that the per-view point clouds that can be recovered from the proposed NeRF architecture contain small-scale wrinkles. Thus, the template mesh can be supervised by a 3D-to-3D constraint between the posed and deformed template and the per-view point cloud, which is explained in more detail in the following.

First, the per-view point cloud for any given ray r (which passes through the dilated/eroded depth map) of the current frame can be computed as

$$\mathbf{p}^{(r)} = \mathbf{o}^{(r)} + \left(\sum_{i=0}^{K} T_i^{(r)} \alpha_i^{(r)} t_i^{(r)}(\mathbf{V}) \right) \mathbf{d}^{(r)}. \tag{6.26}$$

Then, all rays R of the current view, which have an accumulated density that is higher than a threshold T, define the set of points in the point cloud. This point cloud, which contains the higher frequency geometric details, can be used to supervise the underlying template mesh by employing a Chamfer loss

$$\begin{aligned} \mathcal{L}_{\text{chamfer}}(\mathbf{V}) = &\sum_{i=0}^{N} \eta \left(min_{r \in \{0,\dots,R\}} \|\mathbf{V}_i - \mathbf{p}^{(r)}\|^2 \right) \\ &+ \sum_{r=0}^{R} \eta \left(min_{i \in \{0,\dots,N\}} \|\mathbf{p}^{(r)} - \mathbf{V}_i\|^2 \right) \end{aligned} \tag{6.27}$$

where N is the number of template vertices and $\eta(\cdot)$ is a robust loss function that sets the value to zero when it exceeds a certain threshold to ensure a robustness with respect to outliers. Now, DeltaNet can be refined with the losses introduced in Equation 6.15 and in addition with the proposed Chamfer loss. Further, an isometry or edge length constraint is imposed similar to the one proposed by (Habermann et al., 2019). This constraint has the advantage that it allows local rotations in contrast to the Laplacian regularization, which is important when trying to reproduce wrinkle patterns. It can be seen that the Chamfer loss can help to recover finer wrinkles in the geometry, which are hard to learn using only the weakly supervised losses for the aforementioned reasons.

So far, a combined and deep representation of an explicit mesh and an implicit NeRF has been introduced. Next, it is explained how this representation is trained and how the explicit and implicit geometry component iteratively improve each other.

6.12.4 *Supervision and Training Procedure*

After the introduction of the individual components in the previous sections, it is explained how the combined representation can be trained. To this end, it is assumed EGNet and DeltaNet are pretrained as described in Section 6.5 and 6.7, and the posed and deformed template mesh for any frame f can be obtained with the character representation. Again, f is fixed, and the index is dropped in the following.

6.12.4.1 *Training the Geometry-guided NeRF*

In the first stage, the mesh deformation networks are fixed, and only the NeRF network f_{nf} is trained. For readability, it is assumed the ray

r is fixed, and the superscript is dropped. To this end, the network weights are supervised with the loss

$$\mathcal{L}_{\text{nerf}} = \mathcal{L}_{\text{color}} + \mathcal{L}_{\text{mask}} + \lambda_{\text{variance}}\mathcal{L}_{\text{variance}} \tag{6.28}$$

which consists of two image-based data terms and one regularizer where $\lambda_{\text{variance}}$ is a weighting factor. $\mathcal{L}_{\text{color}}$ is the same color term as proposed in the original work (Mildenhall et al., 2020) but applied on the novel NeRF architecture.

Then, a mask loss

$$\mathcal{L}_{\text{mask}}(\alpha_0, ..., \alpha_K) = \left(\left(\sum_{i=0}^{K} T_i \alpha_i \right) - \mathcal{F} \right)^2 \tag{6.29}$$

is employed, which ensures that the accumulated density values of rays are one when they hit the foreground mask $\mathcal{F}$ and zero when they hit the background. By that, sharper boundaries can be ensured. Further, by ensuring the accumulated density is one in the foreground, it is implicitly ensured that the accumulated depth is close to the depth of the deformed and posed mesh. This is due to the fact that the samples are drawn around the mesh, and by ensuring the individual weights sum up to one, the resulting depth estimate is approximately a linear combination of the depth samples.

Further, the skin and the clothing of humans normally are quite Lambertian, and the goal is to train a NeRF such that the accumulated depth is accurate and can be used for refining the mesh deformation networks. Thus, the volume rendering is constrained to have a small variance along the depth in terms of the density values, and they are constrained by the variance loss

$$\mathcal{L}_{\text{variance}}(\alpha_0, ..., \alpha_K) = ((Var(\alpha_0, ..., \alpha_K))^2 \tag{6.30}$$

$$Var(\alpha_0, ..., \alpha_K) = \sum_{j=0}^{K} \alpha_j \left(t_j - \left(\sum_{i=0}^{K} T_i \alpha_i t_i(\mathbf{V}) \right) \right)^2 \tag{6.31}$$

where $Var(\cdot)$ computes the variance along the depth of the ray, and $\mathcal{L}_{\text{variance}}$ ensures that the variance is small by applying an $L2$ loss on the variance.

6.12.4.2 *Refinement of the Template Mesh*

In the second stage, the NeRF network weights are fixed and only used to create the per-view point clouds. Then, DeltaNet is refined based on these point clouds according to Equation 6.26, where random views are sampled per training iteration. The final loss is a combination of the losses used to train DeltaNet of the original DDC approach (see Equation 6.15) and the Chamfer loss proposed in Equation 6.27.

However, the chroma loss is set to zero. After training, the motion-dependent geometry contains higher frequency details due to the better supervision signals, baked-in template wrinkles are removed, and the deformed and posed template better matches the ground truth, as demonstrated in the results.

6.12.4.3 *Iterating the Individual Stages*

Once again, it must be noted that the human surface is actually opaque, and thus the NeRF volume density around the human should converge to a single point in the end. Further, the training stages can be iterated multiple times. The idea here is that once the geometry is refined using the NeRF-based supervision, it can be again used for a better sampling and feature attachment when training NeRF. After this second refinement of the NeRF network, the improved NeRF point clouds can then be used for a better supervision of the template deformations again. This training procedure can be iterated multiple times, and eventually, an end-to-end training of both networks, i.e., DefNet and the NeRF MLP, is also possible at later iterations. Simultaneously, $\lambda_{\text{variance}}$ can be gradually increased during each iteration so that the depth samples of NeRF converge to a real surface instead of a volume, which is along the observation that humans usually have a rather opaque surface.

6.12.5 *Preliminary Results*

6.12.5.1 *Dataset*

To evaluate the proposed approach, the *Vlad* sequence of the DynaCap dataset (Habermann et al., 2021a) is leveraged, which contains 101 camera views. Specifically, for a proof of concept, the method was only trained on multi-view images of a single time step (frame 8510) showing a boxing motion. In addition to the original data, the ground truth geometry is reconstructed using a multi-view stereo approach (*PhotoScan* 2016) that is used to quantitatively compare the proposed approach against the baselines.

6.12.5.2 *Novel View Synthesis on a Single Frame*

First, it is tested whether the mesh-guided NeRF architecture influences the highly accurate synthesis quality of the original NeRF (Mildenhall et al., 2020). To this end, the original NeRF is replaced with the proposed architecture and trained on a single frame. For testing, a camera path orbiting around the static subject is chosen. Here, all camera views are not seen during training. Some example camera views of the proposed method and the original NeRF approach (Mildenhall et al., 2020) are shown in Figure 6.14. Note that the approach can synthesize highly detailed images showing individual cloth wrinkles and also

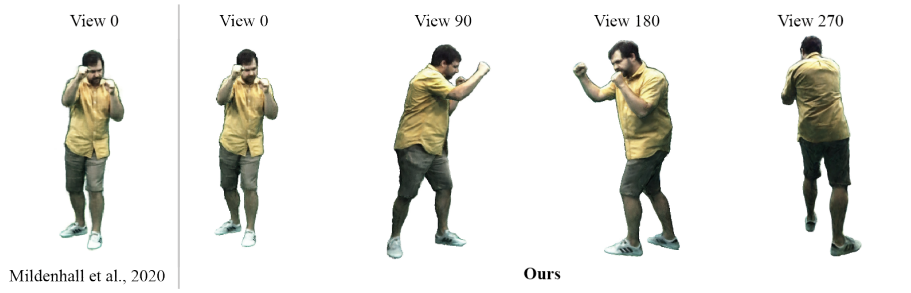
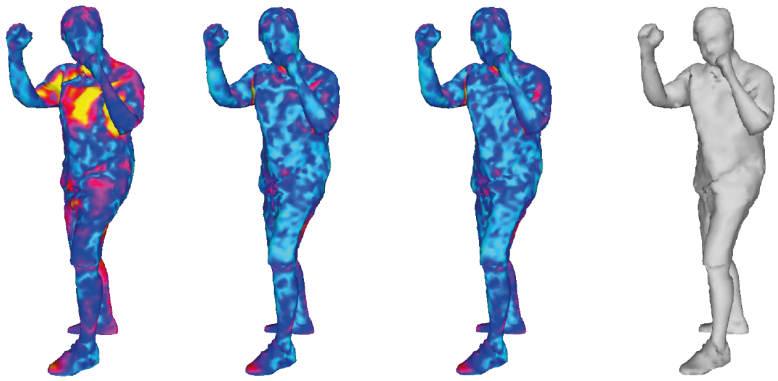


Figure 6.14: Synthesis result on a single frame. Note that also for testing camera views, the method achieves highly accurate and photo-realistic synthesis results. Further, the individual per-view results are also consistent across views. Compared to the original NeRF approach (Mildenhall et al., 2020), the proposed method achieves a comparable synthesis quality while being faster and applicable to dynamic scenes.

smaller textural features such as the ones on the shoes. Further, the appearance of the character is also consistent across individual views. Compared to the original approach, which by design can only handle a static scene, the proposed method still achieves a comparable synthesis quality while the proposed design allows handling dynamic scenes. Thus, the proposed architecture still has the advantages of the original one while potentially being able to handle dynamic scenes. Moreover, the inference speed for a single frame is 14.0 seconds, whereas the original NeRF approach takes 61.9 seconds. This is mainly due to the better sampling, which allows the proposed method to only evaluate rays near the depth maps of the rendered model and the fact that the proposed approach does not require a finer network in addition to the coarse one, as proposed in the original approach.

6.12.5.3 *Geometry Refinement on a Single Frame*

Next, the refinement of the template mesh based on the NeRF-guided supervision is evaluated. To this end, the NeRF trained on the single frame is used, and the per-view point clouds are generated following Equation 6.26. Then, as discussed before, the DeltaNet is refined using the per-view point clouds as supervision. As a baseline, DeltaNet is trained without the point cloud supervision. Both methods are refined for 300 iterations on the single frame. The results are then compared to the ground truth geometry in terms of the per-template-vertex Hausdorff distance averaged over all template vertices. The results are reported in Table 6.6. Using the proposed NeRF-guided supervision clearly improves the result compared to the baseline. Also, more iterations of the proposed iterative scheme further lower the error with respect to the ground truth geometry. Thus, the proposed NeRF-guided supervision drastically improves the geometric accuracy of the template. A visualization of these results is shown in Figure 6.15.



1 Iter. w/o CL 1 Iter. w/ CL 2 Iter. w/ CL 2 Iter. w/ CL

Figure 6.15: Geometry refinement result on a single frame. For all results, DeltaNet is overfitted to the specific frame. The yellow color represents a high error (30mm), and the blue color indicates a low error (0.007mm) in terms of the Hausdorff distance measured between the result and the ground truth scan. Left. The per-vertex error of the posed and deformed template mesh using only the original losses proposed in (Habermann et al., 2021a). Middle. The per-vertex error of the template after the first and second iteration using the NeRF-guided supervision in addition to the losses proposed in (Habermann et al., 2021a). Right. The result after the second iteration using the proposed NeRF-guided supervision. When using the NeRF-guided supervision, the template matches the ground truth much better as wrinkles, and general deformations can be better supervised.

6.12.5.4 *Convergence of NeRF to a Surface*

Next, the iterative refinement is evaluated. To this end, the refined template result is used to guide the NeRF in a second training iteration. In addition, the variance regularizer is now increased as described before to ensure that NeRF converges closer to a surface rather than a volume. In Figure 6.16, a visualization of the NeRF point cloud before and after refinement is shown. One can see that before the first refinement, the recovered point cloud of NeRF is less structured and still contains noise. This is due to the fact that NeRF can compensate for geometric errors by the volume rendering and the view-dependent network branch. However, when refining NeRF with better template guidance and the increased regularization, the NeRF is sampled closer to the true underlying surface, and the increased regularization forces NeRF to learn the true geometry and prevents it from compensating errors by the volume rendering and the view-dependent branch.

6.12.6 *Remaining Challenges*

Although the preliminary results can serve as a proof of concept, there are still remaining challenges to be tackled. Most importantly, it needs to be evaluated if the proposed architecture works well for multiple

1 Iter. w/o CL ↓	1 Iter. w/ CL ↓	2 Iter. w/ CL ↓
12.45mm	7.20mm	**7.11mm**

Table 6.6: Geometric refinement using NeRF point clouds. Here, the refined geometry is compared to the ground truth scan in terms of the Hausdorff distance. First, DeltaNet was refined *without* using the Chamfer loss (CL) and, therefore, the NeRF-guided supervision. Then, DeltaNet was refined by using the Chamfer loss. Clearly, the NeRF-guided supervision helps to better match the ground truth. Last, a second iteration of refinement is performed, which further improves the alignment quality.

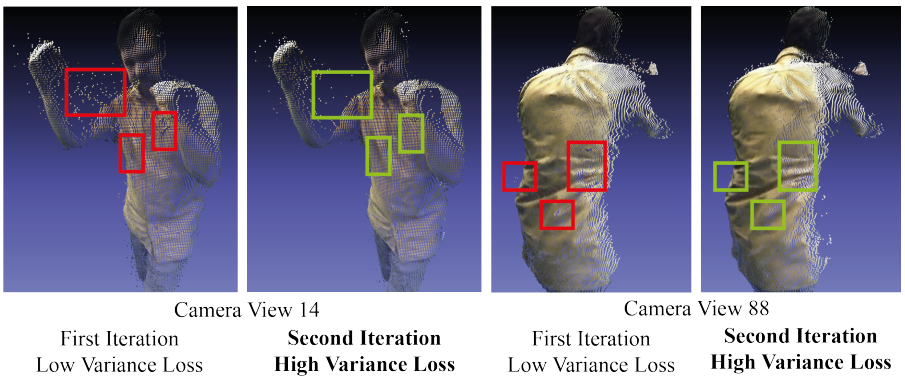


Figure 6.16: Evaluation of the iterative refinement. Left. The NeRF point cloud after the initial training. Right. The NeRF point cloud after training with the already refined mesh and with an increased variance regularization. Note that the result after a second iteration of training is more structured and contains less noise, which indicates that the NeRF better matches the true underlying surface.

frames during training and whether the high quality can be preserved for dynamic scenes. This can also potentially influence the selection of hyperparameters, e.g., the choice of the graph feature size. Moreover, an ablation needs to be performed on the number of training iterations to see when the refinement of each component converged to an optimal state. Another interesting aspect that should be evaluated is the end-to-end training of all components since the sampling procedure is differentiable with respect to the underlying mesh vertices. Further, the generalization of the proposed architecture to novel motions needs to be evaluated. Finally, more subjects and clothing types (especially loose clothing) should be evaluated.

7 CONCLUSION

This thesis has presented new methods for monocular human performance capture, which has further pushed the state of the art in terms of efficiency and 3D accuracy.

LiveCap, presented in Chapter 4, proposes novel algorithmic design choices and a new GPU-based and data-parallel solver architecture in order to achieve real-time performance while capturing the entire dense deforming surface of the human from a single color stream. Chapter 5 presented DeepCap, which is the first learning-based approach for dense, space-time coherent monocular human performance capture that only requires weak multi-view supervision during training. It was shown that this specific supervision can greatly resolve the inherent ambiguities of the monocular setting, and thus it can achieve state-of-the-art 3D accuracy in terms of 3D pose and surface deformation.

However, it is not only important to be able to capture humans in images, but also to be able to synthesize controllable 3D characters and render them into photo-real images from novel viewpoints.

Chapter 6 introduced a novel human synthesis method that allows full control over the motion of the 3D character, the camera position as well as the viewing direction. Moreover, it predicts motion-dependent deforming geometry as well as motion- and view-dependent dynamic textures. The proposed method not only shows photo-real synthesis results for virtual 3D characters but also proofs to be 3D consistent due to the explicit modeling of geometry. This chapter further investigated how a combination of implicit and explicit surface representations can enhance the geometric deformations and the synthesis quality for photo-real humans. Preliminary results show that explicit shape representations can guide the implicit ones and vice versa such that a synergy effect emerges.

7.1 INSIGHTS AND IMPLICATIONS

Beyond the contributions of the presented works in this thesis, there are general insights that can be gained from the individual chapters.

7.1.1 *Image-based Supervision*

One key observation in all presented approaches is that 3D supervision, e.g., in the form of point clouds or registered templates, is not strictly required for capturing the pose and surface deformation or

for learning to synthesize humans. This thesis has shown that 2D losses, which can be based on either sparse joint keypoint detections, foreground silhouettes, or dense photometric losses, in combination with a regularization in the spatial and temporal domain, can remove the need for 3D supervision entirely. This comes with the additional benefit that the number of cameras can be flexible in these approaches, and one can jointly train on multi-view and single-view data. This is to some extent demonstrated in the domain adaptation step of DeepCap, where the same multi-view image-based losses are leveraged but only on a single camera view.

7.1.2 *Coarse-to-fine Modeling and Pose Normalization*

Another insight gained is that when capturing and modeling deformations, the decomposition of deformations into coarse and fine ones is key essential for good capture and synthesis quality. All presented approaches use skinning-based deformation as the coarsest level where the skeletal pose is driving the mesh. In most cases, this leads to a very good initialization for either capturing the remaining deformation or for learning motion-dependent deformations that cannot be explained by skinning. The finer levels of deformation can be consecutively represented by embedded deformation and vertex displacements. Also, 2D displacement or normal maps could be added as proposed in other works (Lähner et al., 2018) to model details beyond the resolution of the mesh. An alternative to these explicit geometry representations for fine-scale details are implicit surface representations that can be combined with an explicit representation, as shown within this thesis (see Section 6.12). Along these lines, it is also shown that it is beneficial for learning-based approaches to take pose normalized inputs when regressing geometric deformations, as this has been shown to simplify the learning for the network and has also demonstrated better generalization to unseen data.

7.1.3 *Regression and Optimization of Model Parameters*

This thesis has presented two monocular human performance capture approaches – one of them based on classical model fitting and one on regression using deep neural architectures. The obvious question is which of these two concepts is the better choice; granted, both approaches have advantages and disadvantages. Model fitting is prone to fall into erroneous local minima, although, when a good solution is found, it can potentially explain the input almost pixel-perfect. Regression typically fails to perfectly match the observation of the input (such as the model overlap onto the input image), but it is usually not prone to fall into local minima. Here, this thesis provides

evidence that a combination of these two concepts can provide the best result. DeepCap, for example, regresses the model parameters from images; however, in the domain adaption step, the network is fine-tuned on the monocular testing data to provide the most accurate result. This step can be viewed as a model fitting using the pre-trained network as initialization which can help to prevent the convergence at a local minimum. Thus, for future projects, a combined approach would be desirable – one that initializes model parameters using regression to avoid converging at a local minimum and applies a final fitting step to better match the image evidence.

7.1.4 *Datasets*

Three novel datasets, which will further stimulate research and serve as benchmarks for future works, were proposed in this thesis. The first dataset is proposed in (Habermann et al., 2019), for which more than 20 minutes of monocular human performances were recorded, including challenging motions, clothing types, and environments. This dataset is ideal for testing monocular human performance capture approaches under in-the-wild conditions. The second dataset is proposed in (Habermann et al., 2020), for which four subjects were recorded performing an extensive set of motions in a sparse multi-camera studio with a green screen. In addition to image annotations that include the 2D pose and the foreground masks, rigged and skinned meshes are provided along with ground truth 3D poses and in-the-wild test sequences. Finally, a multi-camera capture of individual humans, similar to the DeepCap dataset, is provided in Chapter 6. For this dataset, however, a dense camera system with over 100 cameras was used for recording. Acquiring data at such a scale is far from trivial, and there is not a single one of comparable size and quality publicly available, which means that this dataset can stimulate further research and enable new research directions, e.g., in the domain of dynamic scene representation learning from multi-view imagery.

7.2 FUTURE DIRECTIONS

This thesis has presented methods that advance the state of the art in terms of monocular human performance capture and animated 3D character synthesis. Nevertheless, there are still remaining challenges and open questions towards the vision of detailed and fast capture and synthesis of entire humans from ideally sparse visual data, which opens up future work in these areas. Some promising directions for future work are described in the following.

7.2.1 *Incorporating Physics into Monocular Human Performance Capture*

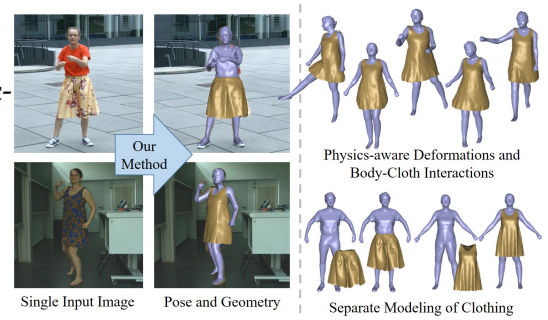


Figure 7.1: Incorporating physics into the learning of the surface deformation of entire humans allows one to model clothing more accurately as a separate piece of geometry. Figure from Li et al., 2020a.

One important aspect currently missing for previous learning-based monocular human performance capture approaches is that the regressed pose and geometry should not only match the image evidence but should also be physically plausible. This means that the 3D character pose should also take physical reality into account, such as gravity, and, on the surface level, clothing should be modeled as separate layers that interact with the underlying driving body of the human. Initial works have shown compelling results for physically plausible body poses (Shimada et al., 2021, 2020) and the modeling of clothing as independent pieces (Li et al., 2020a). However, a unified approach for physically plausible pose and surface deformation that also estimates the physical material parameters has not yet been found and will be an interesting line of research in the future.

7.2.2 *Expressive Full Body Capture*

Figure 7.2: Joint regression of body pose, hand gestures, and facial expressions of a naked human body model. Figure from Zhou et al., 2021.

The work in this thesis has mostly focused on body pose and clothing deformations. However, it is equally important to capture facial expressions, hand gestures, and hair. Capturing hands (Wang et al., 2020a; Zhou et al., 2020) and faces (B R et al., 2021; Kim et al., 2018a; Tewari et al., 2017) are very active research areas, and some works have already started to jointly capture the individual body parts (Joo et al., 2018; Zhou et al., 2021).

An interesting question will be how body pose, clothing, facial expressions, hand gestures, and hair can be reconstructed from monocular image data in *real time* and whether jointly capturing them can help to improve the overall quality, as they can potentially influence each other.

7.2.3 *Different Input Modalities*

Apart from single color images, there are also other types of sensors such as event cameras, and some works have already demonstrated high-speed performance capture with a small memory footprint (Xu et al., 2020). Interesting follow-up could involve mixed data modalities such as RGB and event streams where very high spatial accuracy can be ensured by the RGB camera and high temporal resolution by the event stream. Inertial sensors could also be of interest as a complementary type of supervision with respect to RGB data. Some works already show how the pose can be recovered from a sparse set of sensors (Yi et al., 2021) and how an additional egocentric camera can help to navigate in large 3D scenes (Guzov et al., 2021). However, an interesting question that remains unanswered is whether an *extrinsic* camera and very few inertial sensors or even only a single inertial sensor attached to the captured human can improve the 3D performance without increasing the hardware requirements too much, as regular smartphones already have built-in inertial measurement units.

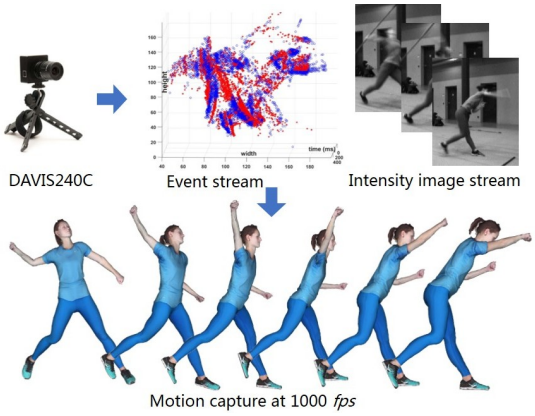


Figure 7.3: High-speed performance capture using a single event camera. Figure from Xu et al., 2020.

7.2.4 *Control over Illumination*

Being able to control the motion of a photo-real character and allowing the change of viewpoint are important steps towards controllable photo-real 3D avatars. However, it is equally important to be able to control and change the scene lighting. Some works (Guo et al., 2019) enable relighting of a captured human performance; but for now, it remains an open question as to how the explicit control of the lighting can be ensured for novel motions. Here, ideas that were proposed

in this thesis could be combined with recent approaches in the area of neural radiance fields (Boss et al., 2020; Mildenhall et al., 2020; Srinivasan et al., 2021) to disentangle the lighting component in the rendering process, allowing it to be directly controlled at inference time.

7.2.5 *Improving the Supervising Loss Functions*

Another promising avenue of future work is to further improve the supervising loss functions. The current rendering losses, used in this thesis, do not account for self shadows and, thus, cannot explain them properly, which can lead to geometric artifacts, e.g., when the geometry tries to explain the shadow. However, some works (Lyu et al., 2021) show fast approximations for shadow computation that could be leveraged to provide a better supervision signal when learning deformations in the context of human performance capture. There are additional properties that would be desirable for rendering losses in the human performance capture context, such as backpropagating gradients into vertices even if they are occluded. These properties would be beneficial because they even allow the supervision of occluded body parts, e.g., when the arm is visible in the ground truth image but not in the current model estimate.

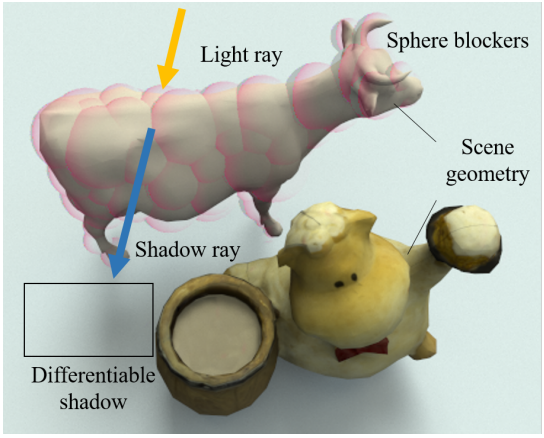


Figure 7.4: Efficient and differentiable computation of visibility and shadows. Figure from Lyu et al., 2021.

7.2.6 *Generalization across Identities*

One limitation of the methods presented in this thesis is that they rely on a pre-scanned template and are, therefore, person-specific. Interesting questions for future research will include whether a mesh representation can be found, which can represent a class of humans and apparel types. Moreover, such a representation should still provide sufficient prior information to eliminate ambiguities in the setups that were dealt with in this thesis, while allowing for the modeling of a wide range of body types and clothing shapes, including, e.g., trousers and skirts. Moreover, the dimensions needing sampling will drastically

increase, which means efficient data capture, processing, and storage will become a relevant problem, as well as the question of how much data is necessary to sufficiently sample the solution space. Finally, it remains to be answered whether it is possible to fully separate identity and pose-dependent aspects, which would be desirable for enabling separate control of the pose and the identity, e.g., when applying new motions on a fixed identity or interpolating the identity in a particular pose.

7.3 FINAL CONCLUSION

Being able to analyze and synthesize humans solely from images can benefit our daily lives in so many ways, from medical applications for better diagnoses through automated motion analysis and immersive communication between people perhaps separated by thousands of kilometers to movie production and games. Thus, earlier work has begun to focus on the emerging problems of the future and has shown tremendous progress in these areas. This thesis has provided key solutions that have further improved human performance capture in terms of run-time performance, 3D accuracy, and have also enabled a synthesis quality never seen before in the context of intuitively controllable 3D avatars. The aim of this thesis is to further stimulate research with the vision of achieving even more realistic capture and synthesis of entire humans, while, at the same time, the hardware setup hopefully gradually becomes simpler such that this technology can be democratized to everyone.

A

APPENDIX

This chapter provides the implementation details of the presented approaches, including, for example, the hyperparameters and the training strategies.

A.1 IMPLEMENTATION DETAILS FOR DEEPCAP (CHAPTER 5)

Both network architectures, as well as the GPU-based custom layers, are implemented in the Tensorflow framework (Abadi et al., 2015). The Adam optimizer (Kingma and Ba, 2014) is used in all experiments.

A.1.1 *Training Strategy for PoseNet*

As one is interested in joint angle regression, one has to note that multiple solutions for the joint angles exist due to the fact that every correct solution can be multiplied by 2π, leading to the same loss value. To this end, training has to be carefully designed. In general, the strategy first focuses on the torso markers by giving them more weight (see Section 5.5). Using this strategy, the global rotation will be roughly correct, and joint angles are slowly trained to avoid overshooting of angular values. This is further ensured by the limits term. After several epochs, when the network already learned to fit the poses roughly, the regularization is turned off, and the angles are further refined. More precisely, the training of *PoseNet* proceeds in three stages. First, *PoseNet* is trained for $120k$ iterations with a learning rate of 10^{-5} and weight $\mathcal{L}_{\text{kp}}$ with 0.01. $\mathcal{L}_{\text{limit}}$ has a weight of 1.0 for the first $40k$ iterations. Between $40k$ and $60k$ iterations, $\mathcal{L}_{\text{limit}}$ is re-weighted with a factor of 0.1. Finally, $\mathcal{L}_{\text{limit}}$ is set to zero for the remaining training steps. Second, *PoseNet* is trained for another $120k$ iterations with a learning rate of 10^{-6} and $\mathcal{L}_{\text{kp}}$ is weighted with a factor of 10^{-4}. Third, *PoseNet* is trained for again $120k$ iterations with a learning rate of 10^{-6} and $\mathcal{L}_{\text{kp}}$ is weighted with a factor of 10^{-5}. A batch size of 90 is always used.

A.1.2 *Training Strategy for DefNet*

DefNet is trained for $120k$ iterations with a batch size of 50. A learning rate of 10^{-5} is used and $\mathcal{L}_{\text{sil}}$, $\mathcal{L}_{\text{kpg}}$, and $\mathcal{L}_{\text{arap}}$ are weighted with $1k$, 0.05, and $1.5k$ respectively.

A.1.3 *Training Strategy for the Domain Adaptation*

To fine-tune the network for in-the-wild monocular test sequences, the pre-trained *PoseNet* and *DefNet* are trained for 250 iterations, respectively. To this end, the multi-view loss is replaced with a single view loss which can be trivially achieved. For *PoseNet*, $\mathcal{L}_{\text{limit}}$ is disabled and $\mathcal{L}_{\text{kp}}$ is weighed with 10^{-6}. For *DefNet*, $\mathcal{L}_{\text{sil}}$, $\mathcal{L}_{\text{kpg}}$, and $\mathcal{L}_{\text{arap}}$ are weighted with $1k$, 0.05, and $1.5k$ respectively. Further, a learning rate of 10^{-6} and the same batch sizes as before are used. This fine-tuning in total takes around 5 minutes.

A.2 IMPLEMENTATION DETAILS FOR DDC (CHAPTER 6)

In all experiments, the Adam optimizer (Kingma and Ba, 2014) is used. Due to the memory limits and training time, 40 cameras views (if available) are randomly sampled for all multi-view losses. The distance transform images have a resolution of 350×350. The rendering resolution of the differentiable renderer is 512×512 (643×470) for the training of DeltaNet and the lighting optimization and 1024×1024 (1285×940) for the training of TexNet. EGNet is trained for 360,000 iterations with a batch size of 40 where the silhouette and ARAP term are balanced with 100.0 and 1500.0, respectively, and a learning rate of 0.0001 is used. This step takes 20 hours using 4 NVIDIA Quadro RTX 8000 with 48GB of memory. The lighting is optimized with a batch size of 4, a learning rate of 0.0001, and 30,000 iterations. This takes around 7 hours. For training DeltaNet, the chroma, silhouette, and Laplacian loss are balanced with 0.03775, 500.0, and 100,000.0, respectively. Again, the network is trained for 360,000 iterations using a batch size of 8 and a learning rate of 0.0001 which takes 2 days. Finally, for training TexNet, a batch size of 12 and a learning rate of 0.0001 are used. 720,000 iterations are applied, which takes 4 days.

A.3 IMPLEMENTATION DETAILS FOR CHAPTER 6.12

t_{mi} and t_{ma} are both set to 4cm. The graph features attached to the vertices have a feature size of 64. The erosion and dilation operator, which are applied to the depth maps, have a stencil size of 9×9. The density threshold T for generating the per-view point clouds is set to 0.98. The distance threshold for the proposed Chamfer loss is set to 4.0cm. The variance weight for the first iteration of training is set to 0.01 and then increased to 0.1 for the second iteration. 64 samples are drawn per ray, and one batch contains 1000 rays during training randomly sampled from one camera view. For the single frame case, the graph convolutional network, which encodes the motion as per-vertex features, has 8 residual blocks with a feature size of 16. The

other hyperparameters are equal to the ones reported for the DeltaNet in Section 6.7. For the first iteration, the NeRF was trained for 110,000 iterations; then, the DeltaNet was fine-tuned for 400 iterations. For the second iteration, NeRF was trained for another 500,000 iterations, and DeltaNet was again fine-tuned for 400 iterations. The weighting terms for the silhouette loss, the laplacian regularizer, the Chamfer loss, and the isometry loss are adjusted for the refinement of DeltaNet, and they are set to 50.0, 4000.0, 5000.0, and 0.075, respectively.

BIBLIOGRAPHY

Abadi, Martín et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org.

Aberman, Kfir, Mingyi Shi, Jing Liao, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or (2019). "Deep Video-Based Performance Cloning." In: *Comput. Graph. Forum* 38.2, pp. 219–233.

Allain, Benjamin, Jean-Sébastien Franco, and Edmond Boyer (2015). "An Efficient Volumetric Framework for Shape Tracking." In: *CVPR 2015 - IEEE International Conference on Computer Vision and Pattern Recognition*. Boston, United States: IEEE, pp. 268–276.

Alldieck, Thiemo, Marcus Magnor, Bharat Lal Bhatnagar, Christian Theobalt, and Gerard Pons-Moll (2019a). "Learning to Reconstruct People in Clothing from a Single RGB Camera." In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1175–1186.

Alldieck, Thiemo, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll (2018a). "Detailed Human Avatars from Monocular Video." In: *International Conference on 3D Vision*, pp. 98–109.

– (2018b). "Video Based Reconstruction of 3D People Models." In: *IEEE Conference on Computer Vision and Pattern Recognition*. CVPR Spotlight Paper.

Alldieck, Thiemo, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor (2019b). "Tex2Shape: Detailed Full Human Body Geometry from a Single Image." In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Anguelov, Dragomir, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis (2005). "SCAPE: Shape Completion and Animation of People." In: *ACM Transactions on Graphics* 24.3, pp. 408–416.

B R, Mallikarjun, Ayush Tewari, Tae-Hyun Oh, Tim Weyrich, Bernd Bickel, Hans-Peter Seidel, Hanspeter Pfister, Wojciech Matusik, Mohamed Elgharib, and Christian Theobalt (2021). *Monocular Reconstruction of Neural Face Reflectance Fields*. arXiv: 2008.10247.

Bailey, Stephen W., Dave Otte, Paul Dilorenzo, and James F. O'Brien (2018). "Fast and Deep Deformation Approximations." In: *ACM Transactions on Graphics* 37.4. Presented at SIGGRAPH 2018, Los Angeles, 119:1–12.

Bălan, Alexandru O and Michael J Black (2008). "The naked truth: Estimating body shape under clothing." In: *European Conference on Computer Vision*. Springer, pp. 15–29.

Balan, Alexandru O, Leonid Sigal, Michael J Black, James E Davis, and Horst W Haussecker (2007). "Detailed human shape and pose from images." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Baran, Ilya and Jovan Popović (2007). "Automatic Rigging and Animation of 3D Characters." In: *ACM Trans. Graph.* 26.3.

Bartoli, A., Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro (2015). "Shape-from-Template." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.10, pp. 2099–2118.

Bhatnagar, Bharat Lal, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll (2019). "Multi-Garment Net: Learning to Dress 3D People from Images." In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Blender (2020). https://www.blender.org/.

Bogo, Federica, Michael J. Black, Matthew Loper, and Javier Romero (2015). "Detailed Full-Body Reconstructions of Moving People from Monocular RGB-D Sequences." In: *International Conference on Computer Vision (ICCV)*, pp. 2300–2308.

Bogo, Federica, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black (2016). "Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image." In: *European Conference on Computer Vision (ECCV)*.

Borgefors, Gunilla (1986). "Distance transformations in digital images." In: *Computer Vision, Graphics, and Image Processing* 34.3, pp. 344 –371.

Boss, Mark, Raphael Braun, Varun Jampani, Jonathan T. Barron, Ce Liu, and Hendrik P.A. Lensch (2020). "NeRD: Neural Reflectance Decomposition from Image Collections." In: *CoRR*.

Bray, Matthieu, Pushmeet Kohli, and Philip HS Torr (2006). "Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts." In: *European conference on computer vision*. Springer, pp. 642–655.

Brox, Thomas, Bodo Rosenhahn, Daniel Cremers, and Hans-Peter Seidel (2006). "High accuracy optical flow serves 3-D pose tracking: exploiting contour and flow based constraints." In: *European Conference on Computer Vision*. Springer, pp. 98–111.

Brox, Thomas, Bodo Rosenhahn, Juergen Gall, and Daniel Cremers (2010). "Combined region and motion-based 3D tracking of rigid and articulated objects." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.3, pp. 402–415.

Caelles, Sergi, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool (2017). "One-Shot Video Object Segmentation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Cagniart, Cedric, Edmond Boyer, and Slobodan Ilic (2010). "Free-form mesh tracking: a patch-based approach." In: *Computer Vision and*

Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, pp. 1339–1346.

Cao, Chen, Derek Bradley, Kun Zhou, and Thabo Beeler (2015). "Real-time High-fidelity Facial Performance Capture." In: *ACM Trans. Graph* 34.4, 46:1–46:9.

Cao, Zhe, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh (2018). "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields." In: *arXiv preprint arXiv:1812.08008*.

Cao, Zhe, Tomas Simon, Shih-En Wei, and Yaser Sheikh (2017). "Real-time Multi-Person 2D Pose Estimation using Part Affinity Fields." In: *CVPR*.

Carranza, Joel, Christian Theobalt, Marcus A. Magnor, and Hans-Peter Seidel (2003). "Free-viewpoint Video of Human Actors." In: *ACM Trans. Graph.* 22.3.

Casas, Dan, Marco Volino, John Collomosse, and Adrian Hilton (2014). "4D Video Textures for Interactive Character Appearance." In: *Comput. Graph. Forum* 33.2.

Chan, Caroline, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros (2019). "Everybody Dance Now." In: *IEEE International Conference on Computer Vision (ICCV)*.

Chen, Wenzheng, Jun Gao, Huan Ling, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler (2019). "Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer." In: *Advances In Neural Information Processing Systems*.

Chen, Xiaowu, Yu Guo, Bin Zhou, and Qinping Zhao (2013). "Deformable model for estimating clothed and naked human shapes from a single image." In: *The Visual Computer* 29.11, pp. 1187–1196.

Chibane, Julian, Thiemo Alldieck, and Gerard Pons-Moll (2020). "Implicit Functions in Feature Space for 3D Shape Reconstruction and Completion." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Choi, Hongsuk, Gyeongsik Moon, and Kyoung Mu Lee (2020). "Pose2Mesh: Graph Convolutional Network for 3D Human Pose and Mesh Recovery from a 2D Human Pose." In: *European Conference on Computer Vision (ECCV)*.

Choi, Kwang-Jin and H. Ko (2005). "Research problems in clothing simulation." In: *Comput. Aided Des.* 37, pp. 585–592.

Collet, Alvaro, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan (2015). "High-quality streamable free-viewpoint video." In: *ACM Transactions on Graphics (TOG)* 34.4, p. 69.

De Aguiar, Edilson, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun (2008). "Performance capture from sparse multi-view video." In: *ACM Transactions on Graphics (TOG)*. Vol. 27. 3. ACM, p. 98.

Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst (2016). "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering." In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., pp. 3844–3852.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). "ImageNet: A Large-Scale Hierarchical Image Database." In: *CVPR09*.

Dou, Mingsong, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi (2017). "Motion2Fusion: Real-time Volumetric Performance Capture." In: *ACM Trans. Graph.* 36.6, 246:1–246:16.

Dou, Mingsong, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. (2016). "Fusion4d: Real-time performance capture of challenging scenes." In: *ACM Transactions on Graphics (TOG)* 35.4, p. 114.

Esser, Patrick, Johannes Haux, Timo Milbich, and Bj orn Ommer (2018). "Towards Learning a Realistic Rendering of Human Behavior." In: *The European Conference on Computer Vision (ECCV) Workshops*.

Feng, Wei-Wen, Yizhou Yu, and Byung-Uck Kim (2010). "A Deformation Transformer for Real-Time Cloth Animation." In: *ACM Trans. Graph.* 29.4.

Gabeur, Valentin, Jean-Sébastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez (2019). "Moulding humans: Non-parametric 3d human shape estimation from single images." In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2232–2241.

Gafni, Guy, Justus Thies, Michael Zollhöfer, and Matthias Nießner (2021). "Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8649–8658.

Gall, Juergen, Carsten Stoll, Edilson De Aguiar, Christian Theobalt, Bodo Rosenhahn, and Hans-Peter Seidel (2009). "Motion capture using joint skeleton tracking and surface estimation." In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, pp. 1746–1753.

Garg, R., A. Roussos, and L. Agapito (2013). "Dense Variational Reconstruction of Non-rigid Surfaces from Monocular Video." In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1272–1279.

Garrido, Pablo, Michael Zollhoefer, Dan Casas, Levi Valgaerts, Kiran Varanasi, Patrick Perez, and Christian Theobalt (2016). "Reconstruction of Personalized 3D Face Rigs from Monocular Video." In: 35.3, 28:1–28:15.

Gong, Ke, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin (2017). "Look Into Person: Self-Supervised Structure-Sensitive

Learning and a New Benchmark for Human Parsing." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Guan, Peng, Loretta Reiss, David A. Hirshberg, Alexander Weiss, and Michael J. Black (2012). "DRAPE: DRessing Any PErson." In: *ACM Trans. Graph.* 31.4.

Guan, Peng, Alexander Weiss, Alexandru O Bălan, and Michael J Black (2009). "Estimating human shape and pose from a single image." In: *ICCV*, pp. 1381–1388.

Güler, Riza Alp, Natalia Neverova, and Iasonas Kokkinos (2018). "DensePose: Dense Human Pose Estimation In The Wild." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gundogdu, Erhan, Victor Constantin, Amrollah Seifoddini, Minh Dang, Mathieu Salzmann, and Pascal Fua (2019). "Garnet: A Two-stream Network for Fast and Accurate 3D Cloth Draping." In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE.

Guo, Kaiwen, Peter Lincoln, Philip Davidson, Jay Busch, Xueming Yu, Matt Whalen, Geoff Harvey, Sergio Orts-Escolano, Rohit Pandey, Jason Dourgarian, Danhang Tang, Anastasia Tkach, Adarsh Kowdle, Emily Cooper, Mingsong Dou, Sean Fanello, Graham Fyffe, Christoph Rhemann, Jonathan Taylor, Paul Debevec, and Shahram Izadi (2019). "The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting." In: *ACM Trans. Graph.* 38.6.

Guo, Kaiwen, Jonathan Taylor, Sean Fanello, Andrea Tagliasacchi, Mingsong Dou, Philip Davidson, Adarsh Kowdle, and Shahram Izadi (2018). "TwinFusion: High Framerate Non-Rigid Fusion through Fast Correspondence Tracking." In:

Guo, Kaiwen, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu (2017). "Real-Time Geometry, Albedo, and Motion Reconstruction Using a Single RGB-D Camera." In: *ACM Transactions on Graphics (TOG)* 36.3, p. 32.

Guo, Yu, Xiaowu Chen, Bin Zhou, and Qinping Zhao (2012). "Clothed and naked human shapes estimation from a single image." In: *Proc. of Computational Visual Media (CVM)*, pp. 43–50.

Guzov, Vladimir, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll (2021). "Human POSEitioning System (HPS): 3D Human Pose Estimation and Self-localization in Large Scenes from Body-Mounted Sensors." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Habermann, Marc, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt (2021a). "Real-time Deep Dynamic Characters." In: *Proceedings of Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*. ACM.

Habermann, Marc, Weipeng Xu, Helge Rhodin, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt (2018). "NRST: Non-rigid Surface Tracking from Monocular Video." In: *Proceedings of the Ger-*

man Conference on Pattern Recognition (GCPR). Vol. 11269. Springer, pp. 335–348.

Habermann, Marc, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt (2020). "DeepCap: Monocular Human Performance Capture Using Weak Supervision." In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

– (2021b). "A Deeper Look into DeepCap." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*. IEEE, pp. 1–1.

Habermann, Marc, Weipeng Xu, Michael Zollhöfer, Gerard Pons-Moll, and Christian Theobalt (2019). "LiveCap: Real-Time Human Performance Capture From Monocular Video." In: *ACM Transactions on Graphics (TOG)*. Vol. 38. 2. ACM, 14:1–14:17.

Hahn, Fabian, Bernhard Thomaszewski, Stelian Coros, Robert W. Sumner, Forrester Cole, Mark Meyer, Tony DeRose, and Markus Gross (2014). "Subspace Clothing Simulation Using Adaptive Bases." In: *ACM Trans. Graph.* 33.4.

Hasler, Nils, Hanno Ackermann, Bodo Rosenhahn, Thorsten Thormählen, and Hans-Peter Seidel (2010). "Multilinear pose and body shape estimation of dressed subjects from image sets." In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pp. 1823–1830.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). "Deep Residual Learning for Image Recognition." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Helten, Thomas, Meinard Muller, Hans-Peter Seidel, and Christian Theobalt (2013). "Real-Time Body Tracking with One Depth Camera and Inertial Sensors." In: *The IEEE International Conference on Computer Vision (ICCV)*.

Hesse, Nikolas, Sergi Pujades, Javier Romero, Michael J. Black, Christoph Bodensteiner, Michael Arens, Ulrich G. Hofmann, Uta Tacke, Mijna Hadders-Algra, Raphael Weinberger, Wolfgang Muller-Felber, and A. Sebastian Schroeder (2018). "Learning an Infant Body Model from RGB-D Data for Accurate Full Body Motion Analysis." In: *Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI)*.

Hilsmann, Anna and Peter Eisert (2009). "Tracking and Retexturing Cloth for Real-Time Virtual Clothing Applications." In: *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques*. MIRAGE '09. Berlin, Heidelberg: Springer-Verlag, pp. 94–105.

Hilsmann, Anna, Philipp Fechteler, Wieland Morgenstern, Wolfgang Paier, Ingo Feldmann, Oliver Schreer, and Peter Eisert (2020). "Going beyond free viewpoint: creating animatable volumetric video of human performances." In: *IET Computer Vision* 14.6.

Huang, C.-H., B. Allain, J.-S. Franco, N. Navab, S. Ilic, and E. Boyer (2016). "Volumetric 3D Tracking by Detection." In: *Proc. CVPR*.

Huang, Yinghao, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V. Gehler, Javier Romero, Ijaz Akhter, and Michael J. Black (2017). "Towards Accurate Marker-less Human Shape and Pose Estimation over Time." In: *International Conference on 3D Vision (3DV)*.

Huang, Zeng, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li (2018). "Deep Volumetric Video From Very Sparse Multi-View Performance Capture." In: *Proceedings of the 15th European Conference on Computer Vision*. Computer Vision Foundation.

Innmann, Matthias, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger (2016). "VolumeDeform: Real-time Volumetric Non-rigid Reconstruction." In:

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros (2017a). "Image-to-Image Translation with Conditional Adversarial Networks." In: *CVPR*.

– (2017b). "Image-to-Image Translation with Conditional Adversarial Networks." In: *CVPR*.

Izadi, Shahram, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. (2011). "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera." In: *Proc. UIST*. ACM, pp. 559–568.

Jain, Arjun, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt (2010). "MovieReshape: Tracking and Reshaping of Humans in Videos." In: *ACM Transactions on Graphics* 29.5.

Jiang, Yue, Dantong Ji, Zhizhong Han, and Matthias Zwicker (2020). "SDFDiff: Differentiable Rendering of Signed Distance Fields for 3D Shape Optimization." In: *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jin, N., Y. Zhu, Z. Geng, and R. Fedkiw (2020). "A Pixel-Based Framework for Data-Driven Clothing." In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Goslar, DEU: Eurographics Association.

Joo, Hanbyul, Tomas Simon, and Yaser Sheikh (2018). "Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies." In: *CoRR* abs/1801.01615.

Kadlecek, Petr, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Krivanek, and Ladislav Kavan (2016). "Reconstructing Personalized Anatomical Models for Physics-based Body Animation." In: *ACM Trans. Graph.* 35.6.

Kanazawa, Angjoo, Michael J. Black, David W. Jacobs, and Jitendra Malik (2018). "End-to-end Recovery of Human Shape and Pose." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, pp. 7122–7131.

Kanazawa, Angjoo, Jason Y. Zhang, Panna Felsen, and Jitendra Malik (2019). "Learning 3D Human Dynamics from Video." In: *Computer Vision and Pattern Regognition (CVPR)*.

Kato, Hiroharu, Yoshitaka Ushiku, and Tatsuya Harada (2018). "Neural 3D Mesh Renderer." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kavan, Ladislav, Steven Collins, Jiří Žára, and Carol O'Sullivan (2007). "Skinning with dual quaternions." In: *Proceedings of the 2007 symposium on Interactive 3D graphics and games*. ACM, pp. 39–46.

Kim, Hyeongwoo, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Nießner, Patrick Pérez, Christian Richardt, Michael Zollöfer, and Christian Theobalt (2018a). "Deep Video Portraits." In: *ACM Transactions on Graphics (TOG)* 37.4, p. 163.

– (2018b). "Deep Video Portraits." In: *ACM Transactions on Graphics (TOG)* 37.4, p. 163.

Kim, Meekyoung, Gerard Pons-Moll, Sergi Pujades, Sungbae Bang, Jinwwok Kim, Michael Black, and Sung-Hee Lee (2017). "Data-Driven Physics for Human Soft Tissue Animation." In: *ACM Transactions on Graphics, (Proc. SIGGRAPH)* 36.4.

Kim, Tae-Yong and Eugene Vendrovsky (2008). "DrivenShape: A Data-Driven Approach for Shape Deformation." In: *ACM SIGGRAPH 2008 Talks*. SIGGRAPH '08. New York, NY, USA: Association for Computing Machinery.

Kingma, Diederik and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization." In: *International Conference on Learning Representations*.

Kolotouros, Nikos, Georgios Pavlakos, and Kostas Daniilidis (2019). "Convolutional Mesh Regression for Single-Image Human Shape Reconstruction." In: *CVPR*.

Kowdle, Adarsh, Christoph Rhemann, Sean Fanello, Andrea Tagliasacchi, Jonathan Taylor, Philip Davidson, Mingsong Dou, Kaiwen Guo, Cem Keskin, Sameh Khamis, David Kim, Danhang Tang, Vladimir Tankovich, Julien Valentin, and Shahram Izadi (2018). "The Need 4 Speed in Real-time Dense Visual Tracking." In: *SIGGRAPH Asia 2018 Technical Papers*. SIGGRAPH Asia '18. Tokyo, Japan: ACM, 220:1–220:14.

Kraevoy, Vladislav, Alla Sheffer, and Michiel van de Panne (2009). "Modeling from contour drawings." In: *Proceedings of the 6th Eurographics Symposium on Sketch-Based interfaces and Modeling*. ACM, pp. 37–44.

Lähner, Z., D. Cremers, and Tony Tung (2018). "DeepWrinkles: Accurate and Realistic Clothing Modeling." In: *ECCV*.

Lambert, J. H. (1760). "Photometria sive de mensure de gratibus luminis, colorum umbrae." In: *Photometria sive de mensure de gratibus luminis, colorum umbrae, Eberhard Klett*.

Lassner, Christoph, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V.Gehler (2017). "Unite the People: Closing the Loop Between 3D and 2D Human Representations." In: *Proc. CVPR*.

Leroy, Vincent, Jean-Sébastien Franco, and Edmond Boyer (2017). "Multi-View Dynamic Shape Refinement Using Local Temporal Integration." In: *IEEE, International Conference on Computer Vision 2017*. Venice, Italy.

Levoy, Marc (1990). "Efficient Ray Tracing of Volume Data." In: *ACM Trans. Graph.* 9.3, pp. 245–261.

Li, Guannan, Yebin Liu, and Qionghai Dai (2014). "Free-viewpoint Video Relighting from Multi-view Sequence Under General Illumination." In: *Mach. Vision Appl.* 25.7, pp. 1737–1746.

Li, Tzu-Mao, Miika Aittala, Frédo Durand, and Jaakko Lehtinen (2018). "Differentiable Monte Carlo Ray Tracing through Edge Sampling." In: *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37.6, 222:1–222:11.

Li, Yue, Marc Habermann, Bernhard Thomaszewski, Stelian Coros, Thabo Beeler, and Christian Theobalt (2020a). "Deep Physics-aware Inference of Cloth Deformation for Monocular Human Performance Capture." In: arXiv: 2011.12866 [cs.CV].

Li, Zhengqi, Simon Niklaus, Noah Snavely, and Oliver Wang (2020b). "Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes." In: *https://arxiv.org/abs/2011.13084*.

Liang, Junbang, Ming C. Lin, and Vladlen Koltun (2019). "Differentiable Cloth Simulation for Inverse Problems." In: *Conference on Neural Information Processing Systems (NeurIPS)*.

Liu, Lingjie, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt (2020a). "Neural Sparse Voxel Fields." In: *NeurIPS*.

Liu, Lingjie, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt (2021). "Neural Actor: Neural Free-view Synthesis of Human Actors with Pose Control." In: *ACM Transactions on Graphics (Proc. ACM SIGGRAPH Asia (conditionally accepted))*. ACM.

Liu, Lingjie, Weipeng Xu, Marc Habermann, Michael Zollhöfer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, and Christian Theobalt (2020b). "Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation." In: *Transactions on Visualization and Computer Graphics (TVCG)*. Vol. PP. IEEE, pp. 1–1.

Liu, Lingjie, Weipeng Xu, Michael Zollhöfer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt (2019a). "Neural Rendering and Reenactment of Human Actor Videos." In: *ACM Transactions on Graphics (TOG)*. Vol. 38. 5. ACM.

Liu, Shichen, Tianye Li, Weikai Chen, and Hao Li (2019b). "Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning." In: *The IEEE International Conference on Computer Vision (ICCV)*.

Liu, Yebin, Carsten Stoll, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt (2011). "Markerless motion capture of interacting characters using multi-view image segmentation." In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, pp. 1249–1256.

Lombardi, Stephen, Jason Saragih, Tomas Simon, and Yaser Sheikh (2018). "Deep appearance models for face rendering." In: *ACM Transactions on Graphics* 37.4.

Lombardi, Stephen, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh (2019). "Neural volumes: Learning dynamic renderable volumes from images." In: *ACM Transactions on Graphics (TOG)* 38.4, p. 65.

Loper, Matthew M. and Michael J. Black (2014). "OpenDR: An Approximate Differentiable Renderer." In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, pp. 154–169.

Loper, Matthew, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black (2015). "SMPL: A Skinned Multi-Person Linear Model." In: *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34.6, 248:1–248:16.

Lyu, Linjie, Marc Habermann, Lingjie Liu, Mallikarjun B R, Ayush Tewari, and Christian Theobalt (2021). "Efficient and Differentiable Shadow Computation for Inverse Problems." In: *In Proceedings of the International Conference on Computer Vision (ICCV)*. IEEE.

Ma, Liqian, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool (2017). "Pose guided person image generation." In: *Advances in Neural Information Processing Systems*, pp. 405–415.

Ma, Liqian, Qianru Sun, Stamatios Georgoulis, Luc van Gool, Bernt Schiele, and Mario Fritz (2018). "Disentangled Person Image Generation." In: *Computer Vision and Pattern Recognition (CVPR)*.

Ma, Qianli, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael Black (2020). "Learning to Dress 3D People in Generative Clothing." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Magnenat-Thalmann, N., A. Laperri'ere, and D. Thalmann (1988). "Joint-Dependent Local Deformations for Hand Animation and Object Grasping." In: *Proceedings of Graphics Interface '88*. GI '88. Edmonton, Alberta, Canada: Canadian Man-Computer Communications Society, pp. 26–33.

Martin-Brualla, Ricardo, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello (2018). "LookinGood: Enhancing Performance Capture with Real-Time Neural Re-Rendering." In: *ACM Trans. Graph.* 37.6.

Matusik, Wojciech, Chris Buehler, Ramesh Raskar, Steven J Gortler, and Leonard McMillan (2000). "Image-based visual hulls." In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., pp. 369–374.

Mehta, Dushyant, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt (2017). "VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera." In: vol. 36. 4.

Metaxas, Dimitris and Demetri Terzopoulos (1993). "Shape and non-rigid motion estimation through physics-based synthesis." In: *IEEE Trans. PAMI* 15.6, pp. 580–591.

Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng (2020). "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis." In: *ECCV*.

Mueller, Claus (1966). "Spherical Harmonics." In: *Spherical Harmonics, Springer*.

Mustafa, Armin, Hansung Kim, Jean-Yves Guillemaut, and Adrian Hilton (2015). "General Dynamic Scene Reconstruction from Multiple View Video." In: *ICCV*.

Mustafa, Armin, Hansung Kim, Jean-Yves Guillemaut, and Adrian Hilton (2016). "Temporally Coherent 4D Reconstruction of Complex Dynamic Scenes." In: *CVPR*, pp. 4660–4669.

Narain, Rahul, Armin Samii, and James F. O'Brien (2012). "Adaptive Anisotropic Remeshing for Cloth Simulation." In: *ACM Transactions on Graphics* 31.6. Proceedings of ACM SIGGRAPH Asia 2012, Singapore, 147:1–10.

Natsume, R., S. Saito, Zeng Huang, Weikai Chen, Chongyang Ma, H. Li, and S. Morishima (2019). "SiCloPe: Silhouette-Based Clothed People." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4475–4485.

Nealen, A., Matthias Müller, Richard Keiser, Eddy Boxerman, and M. Carlson (2005). "Physically based deformable models in computer graphics." In: *Eurographics: State of the Art Report*, pp. 71–94.

Newcombe, Richard A., Dieter Fox, and Steven M. Seitz (2015). "DynamicFusion: Reconstruction and Tracking of Non-Rigid Scenes in Real-Time." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Newcombe, Richard A, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon (2011). "KinectFusion: Real-time dense surface mapping and tracking." In: *Proc. ISMAR*. IEEE, pp. 127–136.

Nimier-David, Merlin, Delio Vicini, Tizian Zeltner, and Wenzel Jakob (2019). "Mitsuba 2: A Retargetable Forward and Inverse Renderer." In: *ACM Trans. Graph.* 38.6.

Orts-Escolano, Sergio, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al. (2016). "Holoportation: Virtual 3D Teleportation in Real-time." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, pp. 741–754.

Park, Keunhong, Utkarsh Sinha, Jonathan Barron, Sofien Bouaziz, Dan Goldman, Steven Seitz, and Ricardo Martin-Brualla (2020). "Deformable Neural Radiance Fields." In: *https://arxiv.org/abs/2011.12948*.

Park, Sang Il and Jessica K Hodgins (2008). "Data-driven modeling of skin and muscle deformation." In: *ACM Transactions on Graphics (TOG)*. Vol. 27. 3. ACM, p. 96.

Patel, Chaitanya, Zhouyingcheng Liao, and Gerard Pons-Moll (2020). "TailorNet: Predicting Clothing in 3D as a Function of Human Pose, Shape and Garment Style." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Pavlakos, Georgios, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black (2019). "Expressive Body Capture: 3D Hands, Face, and Body from a Single Image." In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Pavlakos, Georgios, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis (2018). "Learning to Estimate 3D Human Pose and Shape from a Single Color Image." In: *CVPR*.

Peng, Sida, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou (2021). "Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans." In: *CVPR*.

PhotoScan (2016). http://www.agisoft.com.

Pineda, Juan (1988). "A Parallel Algorithm for Polygon Rasterization." In: *SIGGRAPH Comput. Graph.* 22.4.

Plänkers, Ralf and Pascal Fua (2001). "Tracking and modeling people in video sequences." In: *Computer Vision and Image Understanding* 81.3, pp. 285–302.

Pons-Moll, Gerard, Sergi Pujades, Sonny Hu, and Michael Black (2017). "ClothCap: Seamless 4D Clothing Capture and Retargeting." In: *ACM Transactions on Graphics, (Proc. SIGGRAPH)* 36.4.

Pons-Moll, Gerard, Javier Romero, Naureen Mahmood, and Michael J Black (2015). "Dyna: a model of dynamic human shape in motion." In: *ACM Transactions on Graphics (TOG)* 34.4, p. 120.

Popa, Alin-Ionut, Mihai Zanfir, and Cristian Sminchisescu (2017). "Deep Multitask Architecture for Integrated 2D and 3D Human Sensing." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Prada, Fabián, Misha Kazhdan, Ming Chuang, Alvaro Collet, and Hugues Hoppe (2017). "Spatiotemporal atlas parameterization for evolving meshes." In: *ACM Transactions on Graphics (TOG)* 36.4, p. 58.

Pumarola, Albert, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer (2018). "Unsupervised Person Image Synthesis in Arbitrary Poses." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pumarola, Albert, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer (2021). "D-NeRF: Neural Radiance Fields for Dynamic Scenes." In:

Pumarola, Albert, Jordi Sanchez-Riera, Gary P. T. Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer (2019). "3DPeople: Modeling the Geometry of Dressed Humans." In: *The IEEE International Conference on Computer Vision (ICCV)*.

Raj, Amit, Michael Zollhoefer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi (2021). "PVA: Pixel-aligned Volumetric Avatars." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Reuleaux, Franz (1875). "The kinematics of machinery: Outlines of a theory of machines." In: *Dover Publications*.

Rhodin, Helge, Nadia Robertini, Dan Casas, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt (2016). "General Automatic Human Shape and Motion Capture Using Volumetric Contour Cues." In: *ECCV*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, pp. 509–526.

Robertini, Nadia, Dan Casas, Helge Rhodin, Hans-Peter Seidel, and Christian Theobalt (2016). "Model-based Outdoor Performance Capture." In: *International Conference on Computer Vision (3DV)*.

Rogez, Gregory, Philippe Weinzaepfel, and Cordelia Schmid (2017). "LCR-Net: Localization-Classification-Regression for Human Pose." In: *CVPR 2017-IEEE Conference on Computer Vision & Pattern Recognition*.

Rogge, Lorenz, Felix Klose, Michael Stengel, Martin Eisemann, and Marcus Magnor (2014). "Garment replacement in monocular video sequences." In: *ACM Transactions on Graphics (TOG)* 34.1, p. 6.

Romero, Javier, Dimitrios Tzionas, and Michael J. Black (2017). "Embodied Hands: Modeling and Capturing Hands and Bodies Together." In: *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*. 245:1–245:17 36.6, 245:1–245:17.

Russell, Chris, Rui Yu, and Lourdes Agapito (2014). "Video Pop-up: Monocular 3D Reconstruction of Dynamic Scenes." In: *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, pp. 583–598.

Saito, Shunsuke, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li (2019). "PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization." In: *The IEEE International Conference on Computer Vision (ICCV)*.

Saito, Shunsuke, Tomas Simon, Jason Saragih, and Hanbyul Joo (2020). "PIFuHD: Multi-Level Pixel-Aligned Implicit Function for High-Resolution 3D Human Digitization." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Salzmann, Mathieu and Pascal Fua (2011). "Linear local models for monocular reconstruction of deformable surfaces." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5, pp. 931–944.

Santesteban, Igor, Miguel A. Otaduy, and Dan Casas (2019). "Learning-Based Animation of Clothing for Virtual Try-On." In: *Comput. Graph. Forum* 38, pp. 355–366.

Saragih, J. M., S. Lucey, and J. F. Cohn (2009). "Face alignment through subspace constrained mean-shifts." In: *2009 IEEE 12th International Conference on Computer Vision*, pp. 1034–1041.

Sarkar, Kripasindhu, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt (2020). "Neural Re-Rendering of Humans from a Single Image." In: *European Conference on Computer Vision (ECCV)*.

Sekine, M., K. Sugita, F. Perbet, B. Stenger, and M. Nishiyama (2014). "Virtual Fitting by Single-Shot Body Shape Estimation." In: *Int. Conf. on 3D Body Scanning Technologies*, pp. 406–413.

Shimada, Soshi, Vladislav Golyanik, Weipeng Xu, Patrick Pérez, and Christian Theobalt (2021). "Neural Monocular 3D Human Motion Capture." In: *ACM Transactions on Graphics* 40.4.

Shimada, Soshi, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt (2020). "PhysCap: Physically Plausible Monocular 3D Motion Capture in Real Time." In: *ACM Transactions on Graphics* 39.6.

Shysheya, Aliaksandra, E. Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, K. Iskakov, Aleksei Ivakhnenko, Yury Malkov, I. Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and V. Lempitsky (2019). "Textured Neural Avatars." In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2382–2392.

Si, Chenyang, Wei Wang, Liang Wang, and Tieniu Tan (2018). "Multi-stage Adversarial Losses for Pose-Based Human Image Synthesis." In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Siarohin, Aliaksandr, Enver Sangineto, Stephane Lathuiliere, and Nicu Sebe (2018). "Deformable GANs for Pose-based Human Image Generation." In: *CVPR 2018*.

Sigal, Leonid, Sidharth Bhatia, Stefan Roth, Michael J Black, and Michael Isard (2004). "Tracking loose-limbed people." In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. I–421.

Simon, Tomas, Hanbyul Joo, Iain Matthews, and Yaser Sheikh (2017). "Hand Keypoint Detection in Single Images using Multiview Bootstrapping." In: *CVPR*.

Sitzmann, Vincent, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer (2019a). "DeepVoxels: Learning Persistent 3D Feature Embeddings." In: *Computer Vision and Pattern Recognition (CVPR)*.

Sitzmann, Vincent, Michael Zollhöfer, and Gordon Wetzstein (2019b). "Scene Representation Networks: Continuous 3D-Structure-Aware Neural Scene Representations." In: *Advances in Neural Information Processing Systems*.

Slavcheva, Miroslava, Maximilian Baust, Daniel Cremers, and Slobodan Ilic (2017). "KillingFusion: Non-rigid 3D Reconstruction without Correspondences." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 3. 4, p. 7.

Sminchisescu, Cristian and Bill Triggs (2003). "Kinematic jump processes for monocular 3D human tracking." In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 1. IEEE, pp. I–69.

Sorkine, Olga and Marc Alexa (2007). "As-rigid-as-possible Surface Modeling." In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. SGP '07. Barcelona, Spain: Eurographics Association.

Srinivasan, Pratul P., Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron (2021). "NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis." In: *CVPR*.

Starck, Jonathan and Adrian Hilton (2007). "Surface capture for performance-based animation." In: *IEEE Computer Graphics and Applications* 27.3, pp. 21–31.

Stoll, Carsten, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt (2010). "Video-Based Reconstruction of Animatable Human Characters." In: *ACM Trans. Graph.* 29.6.

Su, Zhaoqi, Weilin Wan, Tao Yu, Lingjie Liu, Lu Fang, Wenping Wang, and Yebin Liu (2020). "MulayCap: Multi-layer Human Performance Capture Using A Monocular Video Camera." In: *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1.

Sumner, Robert W., Johannes Schmid, and Mark Pauly (2007). "Embedded Deformation for Shape Manipulation." In: *ACM Trans. Graph.* 26.3.

Sun, Xiao, Jiaxiang Shang, Shuang Liang, and Yichen Wei (2017). "Compositional Human Pose Regression." In: *ICCV*.

Tagliasacchi, Andrea, Matthias Schroeder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly (2015). "Robust Articulated-ICP for Real-Time Hand Tracking." In: *Computer Graphics Forum (Symposium on Geometry Processing)* 34.5.

Tang, Min, tongtong wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha (2018). "I-Cloth: Incremental Collision Handling for GPU-Based Interactive Cloth Simulation." In: *ACM Trans. Graph.* 37.6.

Tao, Yu, Zerong Zheng, Yuan Zhong, Jianhui Zhao, Dai Quionhai, Gerard Pons-Moll, and Yebin Liu (2019). "SimulCap : Single-View Human Performance Capture with Cloth Simulation." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tekin, Bugra, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua (2016). "Structured Prediction of 3D Human Pose with Deep Neural Networks." In: *British Machine Vision Conference (BMVC)*.

Tekin, Bugra, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua (2017). "Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation." In: *ICCV*. IEEE Computer Society, pp. 3961–3970.

Tewari, Ayush, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian (2017). "MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction." In: *The IEEE International Conference on Computer Vision (ICCV)*.

The Captury (2020). http://www.thecaptury.com/.

Thies, Justus, Michael Zollhöfer, and Matthias Nießner (2019). "Deferred neural rendering: image synthesis using neural textures." In: *ACM Transactions on Graphics* 38.

Tome, Denis, Chris Russell, and Lourdes Agapito (2017). "Lifting from the deep: Convolutional 3d pose estimation from a single image." In: *IEEE Conf. on Computer Vision and Pattern Recognition. Proceedings*.

Treedys (2020). https://www.treedys.com/.

Tretschk, Edgar, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt (2020). "Non-Rigid Neural Radiance Fields: Reconstruction and Novel View Synthesis of a Deforming Scene from Monocular Video." In: *https://arxiv.org/abs/2012.12247*.

Varol, Gül, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid (2018). "BodyNet: Volumetric Inference of 3D Human Body Shapes." In: *ECCV*.

Vlasic, Daniel, Ilya Baran, Wojciech Matusik, and Jovan Popović (2008). "Articulated mesh animation from multi-view silhouettes." In: *ACM Transactions on Graphics (TOG)*. Vol. 27. 3. ACM, p. 97.

Vlasic, Daniel, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popović, Szymon Rusinkiewicz, and Wojciech Matusik (2009). "Dynamic shape capture using multi-view photometric stereo." In: *ACM Transactions on Graphics (TOG)* 28.5, p. 174.

Volino, Marco, Dan Casas, John Collomosse, and Adrian Hilton (2014). "Optimal Representation of Multiple View Video." In: *Proceedings of the British Machine Vision Conference*. BMVA Press.

Wang, Huamin, Florian Hecht, Ravi Ramamoorthi, and James F. O'Brien (2010). "Example-Based Wrinkle Synthesis for Clothing Animation." In: *ACM Trans. Graph.* 29.4.

Wang, Jiayi, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A. Otaduy, Dan Casas, and Christian Theobalt (2020a). "RGB2Hands: Real-Time Tracking of 3D Hand Interactions from Monocular RGB Video." In: *ACM Transactions on Graphics (TOG)* 39.6.

Wang, Ruizhe, Lingyu Wei, Etienne Vouga, Qixing Huang, Duygu Ceylan, Gerard Medioni, and Hao Li (2016). "Capturing Dynamic Textured Surfaces of Moving Targets." In: *Proceedings of the European Conference on Computer Vision (ECCV)*.

Wang, Ting-Chun, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro (2018a). "Video-to-Video Synthesis." In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1152–1164.

Wang, Ting-Chun, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro (2018b). "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs." In: *CVPR*.

Wang, Ziyan, Timur Bagautdinov, Stephen Lombardi, Tomas Simon, Jason Saragih, Jessica Hodgins, and Michael ZollhÃ¶fer (2020b). *Learning Compositional Radiance Fields of Dynamic Human Heads*. arXiv: 2012.09955 [cs.CV].

Waschbüsch, Michael, Stephan Würmlin, Daniel Cotting, Filip Sadlo, and Markus Gross (2005). "Scalable 3D video of dynamic scenes." In: *The Visual Computer* 21.8-10, pp. 629–638.

Wei, Shih-En, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh (2016). "Convolutional Pose Machines." In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wei, X., P. Zhang, and J. Chai (2012). "Accurate Realtime Full-body Motion Capture Using a Single Depth Camera." In: *ACM TOG (Proc. SIGGRAPH Asia)* 31.6, 188:1–188:12.

Weiss, Alexander, David Hirshberg, and Michael J Black (2011). "Home 3D body scans from noisy image and range data." In: *Proc. ICCV*. IEEE, pp. 1951–1958.

Wu, Chenglei, Carsten Stoll, Levi Valgaerts, and Christian Theobalt (2013). "On-set Performance Capture of Multiple Actors With A Stereo Camera." In: *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*. Vol. 32. 6, 161:1–161:11.

Wu, Chenglei, Kiran Varanasi, and Christian Theobalt (2012). "Full body performance capture under uncontrolled and varying illumination: A shading-based approach." In: *ECCV*, pp. 757–770.

Xian, Wenqi, Jia-Bin Huang, Johannes Kopf, and Changil Kim (2020). "Space-time Neural Irradiance Fields for Free-Viewpoint Video." In: *https://arxiv.org/abs/2011.12950*.

Xiang, Donglai, Hanbyul Joo, and Yaser Sheikh (2019). "Monocular total capture: Posing face, body, and hands in the wild." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Xiang, Donglai, Fabian Prada, Chenglei Wu, and Jessica Hodgins (2020). "MonoClothCap: Towards Temporally Coherent Clothing Capture from Monocular RGB Video." In: *Proceedings of International Conference on 3D Vision (3DV '20)*, pp. 322 –332.

Xu, Feng, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt (2011). "Video-based Characters: Creating New Human Performances from a Multi-view Video Database." In: *ACM SIGGRAPH 2011 Papers*. SIGGRAPH '11. New York, NY, USA: ACM, 32:1–32:10.

Xu, Lan, Weipeng Xu, Vladislav Golyanik, Marc Habermann, Lu Fang, and Christian Theobalt (2020). "EventCap: Monocular 3D Capture of High-Speed Human Motions using an Event Camera." In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Xu, Weipeng, Avishek Chatterjee, Michael Zollhöfer, Helge Rhodin, Dushyant Mehta, Hans-Peter Seidel, and Christian Theobalt (2018). "MonoPerfCap: Human Performance Capture From Monocular Video." In: *ACM Trans. Graph.* 37.2, 27:1–27:15.

Xu, Weiwei, Nobuyuki Umentani, Qianwen Chao, Jie Mao, Xiaogang Jin, and Xin Tong (2014). "Sensitivity-Optimized Rigging for Example-Based Real-Time Clothing Synthesis." In: *ACM Trans. Graph.* 33.4.

Yang, Jinlong, Jean-Sébastien Franco, Franck Hétroy-Wheeler, and Stefanie Wuhrer (2016). "Estimation of Human Body Shape in Motion with Wide Clothing." In: *European Conference on Computer Vision 2016*. Amsterdam, Netherlands.

Ye, Genzhi, Yebin Liu, Nils Hasler, Xiangyang Ji, Qionghai Dai, and Christian Theobalt (2012). "Performance capture of interacting characters with handheld kinects." In: *ECCV*. Vol. 7573 LNCS. PART 2, pp. 828–841.

Ye, Mao and Ruigang Yang (2014). "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2345–2352.

Yi, Xinyu, Yuxiao Zhou, and Feng Xu (2021). *TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors*. arXiv: 2105.04605 [cs.GR].

Yoon, Jae Shin, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt (2020). *Pose-Guided Human Animation from a Single Image in the Wild*. arXiv: 2012.03796 [cs.CV].

Yu, Rui, Chris Russell, Neill D. F. Campbell, and Lourdes Agapito (2015). "Direct, Dense, and Deformable: Template-Based Non-Rigid

3D Reconstruction From RGB Video." In: *The IEEE International Conference on Computer Vision (ICCV)*.

Yu, Tao, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu (2017). "BodyFusion: Real-time Capture of Human Motion and Surface Geometry Using a Single Depth Camera." In: *The IEEE International Conference on Computer Vision (ICCV)*. ACM.

Yu, Tao, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu (2018). "DoubleFusion: Real-time Capture of Human Performances with Inner Body Shapes from a Single Depth Sensor." In: *The IEEE International Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE.

Zhang, Chao, Sergi Pujades, Michael Black, and Gerard Pons-Moll (2017). "Detailed, accurate, human shape estimation from clothed 3D scan sequences." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Spotlight.

Zhang, Kai, Gernot Riegler, Noah Snavely, and Vladlen Koltun (2020). "NERF++: Analyzing and Improving Neural Radiance Fields." In: *https://arxiv.org/abs/2010.07492*.

Zhang, Meng, Tuanfeng Y. Wang, Duygu Ceylan, and N. Mitra (2021). "Deep Detail Enhancement for Any Garment." In: *Computer Graphics Forum* 40.

Zhang, Peizhao, Kristin Siu, Jianjie Zhang, C. Karen Liu, and Jinxiang Chai (2014a). "Leveraging Depth Cameras and Wearable Pressure Sensors for Full-body Kinematics and Dynamics Capture." In: *ACM Transactions on Graphics (TOG)* 33.6, p. 14.

Zhang, Qing, Bo Fu, Mao Ye, and Ruigang Yang (2014b). "Quality Dynamic Human Body Modeling Using a Single Low-cost Depth Camera." In: *CVPR*. IEEE, pp. 676–683.

– (2014c). "Quality dynamic human body modeling using a single low-cost depth camera." In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 676–683.

Zheng, Zerong, Tao Yu, Hao Li, Kaiwen Guo, Qionghai Dai, Lu Fang, and Yebin Liu (2018). "HybridFusion: Real-Time Performance Capture Using a Single Depth Sensor and Sparse IMUs." In: *Proceedings of the 15th European Conference on Computer Vision*. Munich, Germany: Computer Vision Foundation.

Zheng, Zerong, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu (2019). "DeepHuman: 3D Human Reconstruction from a Single Image." In: *The IEEE International Conference on Computer Vision (ICCV)*.

Zhi, Tiancheng, Christoph Lassner, Tony Tung, Carsten Stoll, Srinivasa G. Narasimhan, and Minh Vo (2020). "TexMesh: Reconstructing Detailed Human Texture and Geometry from RGB-D Video." In: *Computer Vision – ECCV 2020*. Lecture Notes in Computer Science. Springer International Publishing.

Zhou, Qian-Yi and Vladlen Koltun (2014). "Color map optimization for 3D reconstruction with consumer depth cameras." In: *ACM Transactions on Graphics (TOG)* 33.4, p. 155.

Zhou, Shizhe, Hongbo Fu, Ligang Liu, Daniel Cohen-Or, and Xiaoguang Han (2010). "Parametric reshaping of human bodies in images." In: *ACM Transactions on Graphics (TOG)* 29.4, p. 126.

Zhou, Xiaowei, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis (2016). "Sparseness meets deepness: 3D human pose estimation from monocular video." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4966–4975.

Zhou, Xingyi, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei (2017). "Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 398–407.

Zhou, Yuxiao, Marc Habermann, Ikhsanul Habibie, Ayush Tewari, Christian Theobalt, and Feng Xu (2021). "Monocular Real-time Full Body Capture with Inter-part Correlations." In: *In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Zhou, Yuxiao, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu (2020). "Monocular Real-time Hand Shape and Motion Capture using Multi-modal Data." In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 0–0.

Zitnick, C Lawrence, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski (2004). "High-quality video view interpolation using a layered representation." In: *ACM Transactions on Graphics (TOG)*. Vol. 23. 3. ACM, pp. 600–608.

Zivkovic, Zoran and Ferdinand van der Heijden (2006). "Efficient Adaptive Density Estimation Per Image Pixel for the Task of Background Subtraction." In: *Pattern Recogn. Lett.* 27.7, pp. 773–780.

Zollhöfer, Michael, Matthias Nießner, Shahram Izadi, Christoph Rhemann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger (2014). "Real-time Non-rigid Reconstruction using an RGB-D Camera." In: *ACM Transactions on Graphics (TOG)* 33.4.

Zurdo, J. S., J. P. Brito, and M. A. Otaduy (2013). "Animating Wrinkles by Example on Non-Skinned Cloth." In: *IEEE Transactions on Visualization and Computer Graphics* 19.1, pp. 149–158.